

Gabor Claußnitzer

**Anbindung der Open Source Enterprise-Content-Management
Lösung Alfresco an ein existierendes ERP System der SAP AG und
prototypische Realisierung eines Fallbeispiels**

eingereicht als

BACHELORARBEIT

an der

HOCHSCHULE MITTWEIDA

UNIVERSITY OF APPLIED SCIENCES

Fakultät für Mathematik/Naturwissenschaften/Informatik

Mittweida, 2010

Gabor Claußnitzer

**Anbindung der Open Source Enterprise-Content-Management
Lösung Alfresco an ein existierendes ERP System der SAP AG und
prototypische Realisierung eines Fallbeispiels**

eingereicht als

BACHELORARBEIT

an der

HOCHSCHULE MITTWEIDA

UNIVERSITY OF APPLIED SCIENCES

Fakultät für Mathematik/Naturwissenschaften/Informatik

Mittweida, 2010

Erstprüfer: Prof. Dr.-Ing. Wilfried Schubert

Zweitprüfer: Dr. Norbert Pruß

Vorgelegte Arbeit wurde verteidigt am: 30.09.2010

Bibliographische Beschreibung:

Claußnitzer, Gabor:

Anbindung der Open Source Enterprise-Content-Management Lösung Alfresco an ein existierendes ERP System der SAP AG und prototypische Realisierung eines Fallbeispiels. - 2010. - 65 S. Mittweida, Hochschule Mittweida, Fakultät für Mathematik/Naturwissenschaften/Informatik, Bachelorarbeit 2010

Referat:

Ziel der Bachelorarbeit ist es, die Relevanz von Dokumenten innerhalb von Geschäftsprozessen anhand eines verbreiteten Anwendungsfalls zu verdeutlichen. Dazu wird neben der Thematik Enterprise-Content-Management auch auf die Möglichkeiten einer Anbindung solcher Systeme an Unternehmenslösungen eingegangen. Im Vordergrund steht dabei eine effiziente Integration von Dokumentenmanagement in bestehende Arbeitsabläufe.

Inhaltsverzeichnis

i.	Abbildungsverzeichnis	6
ii.	Tabellenverzeichnis	7
iii.	Listingsverzeichnis	8
iv.	Abkürzungsverzeichnis	9
0.	Einleitung.....	10
1.	Aufgabenstellung	11
1.1.	Motivation	11
1.2.	Zielsetzung	11
1.3.	Fallbeispiel.....	11
1.4.	Abgrenzung	12
1.5.	Aufbau	12
2.	Enterprise-Content-Management (ECM)	13
2.1.	Definition.....	13
2.2.	Bisherige Entwicklung	14
2.3.	Basis-Komponenten	14
2.4.	Architektur eines ECM-Systems	17
2.5.	Vorteile gegenüber herkömmlicher Datenhaltung.....	18
3.	Referenzlösung Alfresco	20
3.1.	Total Cost of Ownership	20
3.2.	Enterprise Application Integration (EAI).....	21
3.3.	Einsatz von Frameworks	22
3.4.	Systemarchitektur.....	22
3.5.	Anpassbarkeit an spezielle Bedürfnisse	24
3.5.1.	Der Open Source Weg.....	24
3.5.2.	Content-Modelling.....	24
3.5.3.	Konfiguration des Systems	25
4.	Dokumente im Unternehmen.....	26
4.1.	Von der Vision zur Realität	26
4.2.	Bedeutung innerhalb von Geschäftsprozessen	26
4.2.1.	Was ist ein Geschäftsprozess?	26
4.2.2.	Was ist Workflow-Management?	27
4.2.3.	Geschäftsprozesse und Dokumente	27
4.2.4.	Rechtliche Aspekte	29
4.3.	Archivierung als Anwendungsfall.....	30
4.3.1.	Anforderungen	30
4.3.2.	Einsatzziele.....	31
4.3.3.	Vorteile gegenüber Produktivsystemen ohne Archiv.....	31
4.4.	ECM eine vielseitige Lösung	32

5. Dokumentenmanagement mit SAP	33
5.1. Überblick.....	33
5.2. Plattform „NetWeaver“	34
5.3. Relevante Business Services	35
5.4. SAP ArchiveLink.....	36
5.4.1. Verwaltungskonzept.....	37
5.4.2. Ablagearten.....	38
5.5. SAP HTTP Content Server 4.5 Schnittstelle.....	38
5.5.1. Content-Modell.....	39
5.5.2. Arbeitsweise.....	40
5.5.3. Funktionen	40
6. SAP-Alfresco-Connector	42
6.1. Analyse.....	42
6.1.1. Ziel / Produktzweck.....	42
6.1.2. Produkteinsatz	43
6.1.3. Identifikation der Schnittstellen	43
6.1.4. Produktübersicht / Architektur	45
6.1.5. Anwendungsfall „Dokument erzeugen“	46
6.2. Entwurf	47
6.2.1. Charakteristik des Connectors	47
6.2.2. Architekturmodell	48
6.2.3. Verwaltung von Content-Repositoryn	48
6.2.4. Umsetzung des SAP-internen Dokumentenbegriffs.....	49
6.2.5. Sequenz „Dokument erzeugen“	51
6.3. Implementierung.....	52
6.3.1. Einsatz von JAX-RS.....	52
6.3.2. Webservice „Dokument erzeugen“.....	54
6.3.3. Archive-Type-Mapping.....	56
6.3.4. Aufgabe „Dokument spezifizieren“	60
7. Zusammenfassung.....	61
7.1. Fazit.....	61
7.2. Ausblicke	61
8. Literaturverzeichnis	63
v. Erklärung zur selbstständigen Anfertigung	65

i. Abbildungsverzeichnis

<i>Abbildung 1: ECM-Zyklus</i>	<i>16</i>
<i>Abbildung 2: Schichtenmodell</i>	<i>18</i>
<i>Abbildung 3: Integrationstopologie - Hub & Spoke am Beispiel der angestrebten Systemlandschaft</i>	<i>21</i>
<i>Abbildung 4: Alfresco Systemarchitektur, (Quelle: Alfresco Wiki, Alfresco Repository Architecture, 11.05.2009)</i>	<i>23</i>
<i>Abbildung 5: Flussdiagramm: Urlaubsbeantragung</i>	<i>28</i>
<i>Abbildung 6: Flussdiagramm: Anfrage eines Kunden</i>	<i>29</i>
<i>Abbildung 7: Überblick Dokumentenmanagement mit SAP (Quelle: Heck, Rinaldo, Geschäftsprozessorientiertes DMS mit SAP, 2009 S. 33)</i>	<i>33</i>
<i>Abbildung 8: Relevante DMS-Komponenten</i>	<i>36</i>
<i>Abbildung 9: Verknüpfungsdaten (Quelle: SAP AG, SAP ArchiveLink, 2001 S. 54).</i>	<i>38</i>
<i>Abbildung 10: Dokumentenbegriff in SAP (Quelle: SAP AG, SAP ArchiveLink, 2001 S. 239)</i>	<i>40</i>
<i>Abbildung 11: Komponentendiagramm: Architektur des Connectors</i>	<i>46</i>
<i>Abbildung 12: Anwendungsfalldiagramm: Dokument erzeugen</i>	<i>47</i>
<i>Abbildung 13: Klassendiagramm: Content-Repository</i>	<i>49</i>
<i>Abbildung 14: Klassendiagramm: Content-Modell</i>	<i>50</i>
<i>Abbildung 15: Sequenzdiagramm: Dokument erzeugen</i>	<i>51</i>
<i>Abbildung 16: Komponentendiagramm: Redirection</i>	<i>53</i>

ii. Tabellenverzeichnis

<i>Tabelle 1: Arten von Content (Quelle: Kampffmeyer, Enterprise-Content-Management zwischen Vision und Realität, 2003 S. 6).....</i>	<i>13</i>
<i>Tabelle 2: Funktionsbereiche des NetWeaver Application Servers</i>	<i>34</i>
<i>Tabelle 3: Content Server: Funktionsüberblick.....</i>	<i>40</i>
<i>Tabelle 4: Kriterien des SAP-Alfresco Connectors</i>	<i>42</i>

iii. Listingsverzeichnis

<i>Listing 1: Pfad-Annotierung</i>	<i>54</i>
<i>Listing 2: Methodenkopf.....</i>	<i>54</i>
<i>Listing 3: Repository erlangen</i>	<i>55</i>
<i>Listing 4: Multipart Form-Data erzeugen.....</i>	<i>55</i>
<i>Listing 5: Dokument erstellen</i>	<i>55</i>
<i>Listing 6: Komponenten erstellen</i>	<i>56</i>
<i>Listing 7: Antwort generieren.....</i>	<i>56</i>
<i>Listing 8: CMIS-Typdefinition.....</i>	<i>58</i>
<i>Listing 9: SAP-Funktionsbaustein.....</i>	<i>58</i>
<i>Listing 10: SAP-Tabellenabfrage.....</i>	<i>59</i>
<i>Listing 11: Type-Mapping</i>	<i>59</i>

iv. Abkürzungsverzeichnis

ABAP	Advanced Business Application Programing
CMIS	Content-Management-Interoperability-Services
CMS	Content-Management-System
DMS	Dokumentenmanagementsystem
EAI	Enterprise Application Integration
ECM	Enterprise-Content-Management
ERP	Enterprise Resource Planning
JAXB	Java Architecture for XML Binding
JAX-RS	Java API for RESTful Web Services
JCO	SAP Java Connector
JMX	Java Management Extensions

0. Einleitung

Der Einsatz von betrieblicher Standardsoftware spielt innerhalb der meisten mittelständischen und großen Unternehmen bei der Abbildung von Geschäftsprozessen oder Organisationsstrukturen eine entscheidende Rolle. Was soweit führt, dass ein Tagesgeschäft ohne den Einsatz von Softwarelösungen nahezu undenkbar ist. Trotz der gewachsenen Bedeutung solcher Produkte handelt es sich immer noch um Standardsoftware, welche charakteristisch nur einen klar definierten Anwendungsbereich abdeckt und somit nicht alle individuellen Bedürfnisse des Kunden erfüllt. Um Insellösungen zu vermeiden, existieren vom Hersteller integrierte Schnittstellen. Oftmals passen diese jedoch nur zu kostenintensiven, zusätzlichen Komponenten, die wiederum nur einen Teil der gestellten Anforderungen befriedigen.

Die Betrachtung dieses Sachverhaltes im Zusammenhang mit der Thematik *Enterprise-Content-Management (ECM)* zeigt, dass betriebliche Standardsoftware nur einen Teil der Gesamtanforderungen, meist nur das klassische Content-Management, liefert. Entsprechende Bestandteile wie Dokumentenmanagement, Wissensmanagement, Kollaboration oder Schriftgutverwaltung sind entweder bloß rudimentär oder überhaupt nicht vorhanden und verdeutlichen, wie weit ein solches Produkt von einem eigentlichen ECM-System entfernt ist. Zwar lassen sich durch sogenanntes Customizing¹ eine Reihe von Anpassungen vornehmen, doch spezielle Bedürfnisse und Fähigkeiten bleiben so gut wie immer auf der Strecke oder erliegen den zahlreichen Vorteilen der Standardsoftware wie z.B. dem hohen Verbreitungs- und Reifegrad.

Um die Lücke zwischen Ist- und Soll-Zustand zu schließen, ist es notwendig die fehlenden Aspekte im Rahmen von Individualentwicklungen zu realisieren. Dabei stehen den Entwicklern verschiedene Alternativen zur Verfügung. Eine verbreitete Variante ist dabei der Einsatz von standardisierten Schnittstellen, welche es erlauben Entwicklungen möglichst versionssicher und produktunabhängig zu platzieren.

¹ Die Anpassung des Softwareproduktes an individuelle Bedürfnisse.

1. Aufgabenstellung

Dieser Abschnitt soll an die konkrete Aufgabe heranzuführen und die Rahmenbedingungen dieser Bachelorarbeit festlegen.

1.1. Motivation

Seit einigen Jahren setzen Unternehmen verstärkt auf die Verwendung von Open Source Software statt herkömmlicher proprietärer Lösungen. Vor allem im Bereich der ECM-Systeme etablieren sich immer mehr Produkte, welche den Entscheidungsprozess bei der Auswahl von Software wesentlich beeinflussen. Eines davon beschreibt die Lösung von Alfresco, welche basierend auf einer modularen Java-Architektur eine Verbindung neuester Konzepte und Modelle mit modernsten Technologien aufweist. Da viele bereits gewachsene Systemlandschaften vorwiegend auf Standardsoftware vertrauen, ist die Idee entstanden eine Anbindung zu realisieren, welche es vorsieht Alfresco als Archivkomponente für ein verbreitetes ERP System der SAP AG einzusetzen.

1.2. Zielsetzung

Das Ziel dieser Bachelorarbeit soll die Entwicklung eines Connectors² sein, der als Bindeglied zwischen einem SAP ERP System und einer Alfresco Instanz fungiert. Das spätere Softwareprodukt soll es ermöglichen innerhalb von SAP archivierte Dokumente im Content-Repository von Alfresco abzulegen und bei Bedarf wiederzuerlangen. Des Weiteren soll die Notwendigkeit von Dokumenten innerhalb von Unternehmen in den Zusammenhang mit der Thematik Enterprise-Content-Management gebracht werden, um die Einsatzmöglichkeiten solcher Systeme zu identifizieren.

1.3. Fallbeispiel

Durch Customizing und die Verwendung des SAP-Alfresco-Connectors soll anhand eines Fallbeispiels gezeigt werden, dass vergleichbare Aktivitäten generell realisierbar sind. Der zu verwirklichende Anwendungsfall soll die Archivierung der im operativen System entstehenden Ausgangsrechnungen betreffen.

² Eine Schnittstelle zwischen zwei oder mehreren Softwaresystemen.

1.4. Abgrenzung

Die folgenden Punkte sind nicht Bestandteil dieser Arbeit:

- Es soll keine detaillierte Untersuchung der vom späteren Softwaresystem verwendeten Schnittstellen erfolgen.
- Da eine Beschreibung konkreter Vorgehensweisen zur Integration von Dokumenten in bestehende Arbeitsläufe zu umfangreich und zu speziell wäre, soll mit ECM lediglich eine allgemeine Lösungsvariante aufgezeigt werden.
- Um den Rahmen dieser Bachelorarbeit einzuhalten, soll keine vollständige Abhandlung des Softwareentwicklungsprozesses erstellt, sondern die Realisierung des SAP-Alfresco-Connectors anhand ausgewählter Schwerpunkte behandelt werden.

1.5. Aufbau

Für einen besseren Überblick soll im Folgenden Aufbau sowie Inhalt der einzelnen Kapitel vorgestellt werden.

Kapitel 2 führt in die Thematik Enterprise-Content-Management ein. Dabei werden die Begrifflichkeiten, Bestandteile und Konzepte erläutert, die den Einsatz von ECM in einem Unternehmen kennzeichnen.

In **Kapitel 3** wird die Referenzlösung Alfresco vorgestellt. Anhand der Charakteristika dieser Lösung werden Kriterien aufgezeigt, welche für einen Einsatz im Unternehmen sprechen.

Kapitel 4 erläutert die Notwendigkeit von Dokumenten innerhalb von Geschäftsprozessen. Hierbei wird Archivierung als einer von vielen Anwendungsfällen behandelt.

Kapitel 5 befasst sich mit den für die Realisierung relevanten Bestandteilen des SAP-Dokumentenmanagements. Dies umfasst die Softwarekomponenten, welche eine Anbindung von externen Ablagesystemen ermöglichen.

Kapitel 6 beschreibt die Entwicklung des Connectors. Dabei werden die Schwerpunkte der einzelnen softwaretechnischen Phasen aufgezeigt.

2. Enterprise-Content-Management (ECM)

Im folgenden Kapitel soll eine Einführung in die Thematik ECM erfolgen.

2.1. Definition

Gemäß der Association for Information and Image Management (AIIM) lässt sich der Begriff ECM wie folgt definieren:

„Enterprise-Content-Management umfasst die Technologien zur Erfassung, Verwaltung, Speicherung, Bewahrung und Bereitstellung von Inhalten und Dokumenten zur Unterstützung organisatorischer Prozesse.“ [1]

Betrachtet man diese Definition genauer, so erschließen sich weitere Fragen und es lässt sich erahnen, dass das Thema ECM erheblich komplexer ist. Im Abschnitt **2.3** sollen deshalb die konventionellen Informationstechnologien unter der Überschrift „Basis-Komponenten“ veranschaulicht werden.

Zum Verständnis des Begriffs „Inhalt“ ist es wichtig, ihn noch einmal genauer zu spezifizieren, um die Unterschiede zwischen Inhalten und Dokumenten zu verdeutlichen.

Unter dem Term Inhalt, oftmals auch Content genannt, verbirgt sich eine Trennung zwischen den beschreibenden Daten und dem eigentlichen Inhalt. Im Gegensatz dazu beschreibt ein Dokument eine Einheit, welche meistens an einen bestimmten Zweck und eine Verwendung gebunden ist. Als Content können wiederum alle Arten von unstrukturierten, schwach strukturierten und strukturierten Informationen, die in elektronischer Form zur Nutzung bereitstehen, bezeichnet werden.

Tabelle 1: Arten von Content (Quelle: Kampffmeyer, Enterprise-Content-Management zwischen Vision und Realität, 2003 S. 6)

Strukturierte Inhalte	Daten, die in einem standardisierten Layout aus datenbankgestützten Systemen bereitgestellt werden z.B. formierte Datensätze aus einer Datenbank.
Schwach strukturierte Inhalte	Informationen und Dokumente, die zum Teil Layout und Metadaten mit sich tragen, jedoch nicht standardisiert sind z.B. Textverarbeitungsdateien.
Unstrukturierte Inhalte	Beliebige Informationsobjekte, deren Inhalt nicht

	direkt erschlossen werden kann sowie keine Trennung zwischen Inhalt, Layout und Metadaten besitzen z.B. Bilder, Video- oder Sprachdateien.
--	--

Die Aufgabe eines Enterprise-Content-Management-Systems besteht nun darin, den verschiedenen Inhalten durch Metadaten, Kategorisierung, Ablageort sowie durch Schaffung von Beziehungen zu anderem Content einen Kontext zu verleihen. Dies soll es dem Anwender ermöglichen, gewünschte Inhalte innerhalb kürzester Zeit wiederzufinden und mit Hilfe der verschiedenen Informationstechnologien den Lebenszyklus des Contents maßgeblich zu beeinflussen.

2.2. *Bisherige Entwicklung*

Die Ursprünge von ECM lassen sich bis in die 1980er Jahre zurückführen als die ersten Unternehmen begannen elektronische Dokumente datenbankgestützt zu verwalten. Damalige Anwendungen besaßen noch vorwiegend Client-Server-Architekturen. Dies verschaffte den Inhalten zwar unternehmensweite Verfügbarkeit, produzierte jedoch enorme Kosten, was die Wartung, Anpassbarkeit und Weiterentwicklung betraf. Erst 2001 als Dokumenten- und Web-Content-Management miteinander verwachsen, konnten ganz neue Konzepte und Verfahren, die nun in Enterprise-Content-Management inbegriffen sind, erarbeitet werden. Mit dem Web 2.0 Hype im Jahre 2003 und den damit verbunden Technologien gelang es letztendlich ECM zu etablieren und einen eigenen Markt dafür zu schaffen [2] S. 6.

2.3. *Basis-Komponenten*

Die Basiskomponenten verinnerlichen die bereits in der Definition genannten Technologien und übernehmen damit alle wesentlichen Arbeitsfelder, welche mit Enterprise-Content-Management verbunden werden. Der Theorie zu Folge existieren die folgenden fünf Funktionsbereiche:

- Die *Erfassungs-Komponente* beinhaltet Funktionalität zur Erstellung, Erfassung, Aufbereitung und Verarbeitung von analogen und digitalen Informationen, so dass diese im Anschluss den anderen Komponenten und letztendlich dem Nutzer zur Verfügung stehen.

- Bevor eine Verwaltung von Content erfolgen kann, muss eine temporäre Sicherung der Daten erfolgen. Die *Speicherungs-Komponente* realisiert dies durch die Verwendung unterschiedlichster Repositorien, wie z.B. Dateisysteme, Datenbanken oder Data Warehouses, und ist dabei nicht mit der langfristigen Bewahrung von Dokumenten zu verwechseln.
- Einen sehr komplexen Teil beschreibt die *Verwaltungs-Komponente*, welche sich wiederum in fünf Unterkomponenten gliedern lässt. Durch die damit verbundenen Anwendungen wird eine Bearbeitung, Nutzung und Verwaltung der Inhalte ermöglicht. Beispiele hierfür sind das Definieren von Arbeitsabläufen, das Wiederauffinden von Dokumenten sowie das Setzen von Berechtigungen.
- Die *Bereitstellungs-Komponente* fungiert als sogenannter „Output-Manager“, welcher mit Hilfe von spezifischen Transformationstechniken die Generierung von benutzerspezifischen Dokumenten, unter anderem in Form von HTML-, XML- und PDF-Dokumenten, gewährleistet. Der Endanwender kann die gewünschten Informationen so zu jeder Zeit auf bevorzugter Art und Weise, ob per Internet, E-Mail oder über vorhandene E-Business-Portale, erhalten.
- Die *Bewahrungs-Komponente* ist im Gegensatz zur *Speicherungs-Komponente* für die Langzeitarchivierung von Dokumenten zuständig. Des Weiteren beinhaltet dies auch die Verwaltung von Bewahrungsregeln und das Anbinden von speziellen Speichermedien, welche die Inhalte vor Veränderungen schützen sollen.

Die folgende Abbildung zeigt das Zusammenspiel der einzelnen Komponenten in Form eines Zyklus, der für jedes zu verwaltende Dokument einmal abläuft. Ein Durchlaufen aller Komponenten ist dabei nicht zwangsläufig erforderlich.

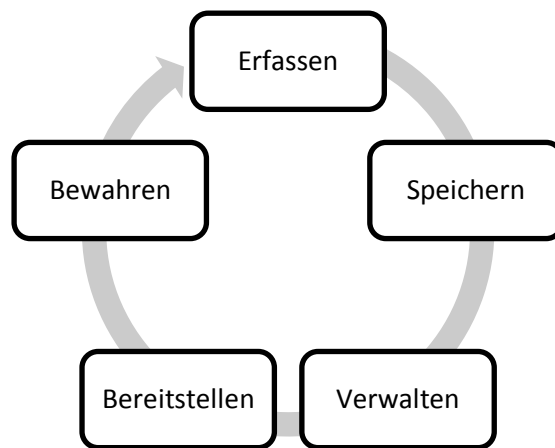


Abbildung 1: ECM-Zyklus

Wie bereits erwähnt, gliedert sich die Verwaltungs-Komponente in mehrere Subkomponenten. Laut dem amerikanischen Forschungsunternehmen Forrester³ lässt sie sich in fünf Hauptanwendungsbereiche unterteilen, welche für ein Unternehmen je nach Einsatzgebiet des Softwareprodukts von ganz unterschiedlicher Relevanz sein können. Eine Aufteilung erfolgt in:

- *Dokumentenmanagement* bezeichnet eine bereits ausgereifte Informationstechnologie, die es ermöglicht physische Dokumente in einer virtuellen Umgebung darzustellen und zu verwalten. Das Aufgabenfeld erstreckt sich über die Erfassung und Verarbeitung von Dokumenten bis hin zur Bereitstellung und Verbreitung.
- *Web-Content-Management* befasst sich mit der Organisation und der Verwaltung von Webseiten und dem dazugehörigen Inhalt.
- *Records-Management* beinhaltet die dauerhafte Aufbewahrung und Archivierung von Schriftgut sowie die systematische Aufzeichnung von Geschäftsvorgängen.
- *Image-Management* beschreibt die Vorgänge zur Verwaltung eingescannter Dokumente. Dazu gehören neben dem Prozess des Scannens, Qualitätskontrolle, Metadatenerfassung sowie die Speicherung von elektronischen Informationen.

³ Ein unabhängiges Technologie- und Marktforschungsunternehmen.

- *Digital-Asset-Management* wird vorrangig in der Marketingbranche eingesetzt. Es befasst sich mit der Erfassung, Erzeugung und Bearbeitung von elektronischen Medien wie z.B. Fotos, Videos und Illustrationen.

Im Gegensatz dazu beinhalten andere Untergliederungen der Verwaltungs-Komponente an Stelle des Image-Managements, welches auch den Erfassungstechnologien zugeordnet werden kann, und dem speziellen Digital-Asset-Management die beiden folgenden Teilkomponenten:

- Mit Hilfe von *Kollaboration* wird eine Unterstützung der Zusammenarbeit einer Gruppe von Personen realisiert. Neben der Überwindung von Hindernissen wie räumlicher und zeitlicher Distanz, ist das Ziel sogenannter Groupware-Applikationen die Bereitstellung eines formalen und kontrollierbaren gemeinsamen Arbeitsbereiches.
- *Workflow-Management* befasst sich mit den Aufgaben der Analyse, Durchführung und Steuerung von Geschäftsprozessen. Es stellt ein Werkzeug in der Form eines Programmes dar, welches eine Verwaltung der einzelnen Workflows ermöglicht. Eine genaue Begriffsdefinition findet sich in Abschnitt 4.2.2.

2.4. *Architektur eines ECM-Systems*

Um den Anforderungen an heutige ECM Lösungen gerecht zu werden, hat sich beim Entwurf gezeigt, dass eine Schichtenarchitektur, sogenanntes Layering, sich als am geeignetsten erweist, wenn es darum geht die zahlreichen Komponenten eines solchen Systems zu organisieren. Eine Trennung des Gesamtsystems erfolgt in fünf Schichten. Dabei ist zu beachten, dass ein Zugriff lediglich von einer „höher“ gelegenen Schicht auf eine „tiefer“ liegende erfolgen kann. Die folgende Abbildung soll dies verdeutlichen:



Abbildung 2: Schichtenmodell

Die *Anwendungsschicht* beinhaltet die in Punkt **2.3** genannten Komponenten eines Enterprise-Content-Management-Systems und bildet mit dem darunterliegenden Repositorium den Kern der Anwendung. Schnittstellen, welche die Grundlage schaffen ein Softwareprodukt in die vorhandene Systemlandschaft zu integrieren, setzen direkt auf das Content-Repository und dessen Dienste auf. Externe Anwendungen erhalten somit einen Zugang zur *Datenzugriffsschicht* und damit auf verwaltete Inhalte sowie deren Metadaten. Durch den Einsatz dieses Architekturmodells ist es möglich einzelne Schichten wie z.B. *Präsentation* auszutauschen ohne die darunter liegende Logik zu beeinflussen.

2.5. Vorteile gegenüber herkömmlicher Datenhaltung

Bei der Entwicklung von Softwaresystemen jeglicher Art stößt man früher oder später immer wieder auf das Problem der Datenhaltung. Ein Großteil der Anwendungen wählt dabei einen der beiden herkömmlichen Wege, welche im folgenden Abschnitt erläutert werden sollen.

Eine Vorgehensweise beinhaltet die Nutzung des klassischen Dateisystems. Dabei werden Routinen und Werkzeuge des Betriebssystems genutzt, um die anwendungsspezifischen Dateien zu bewahren. Da es keinen Kontext gibt, in dem die verwalteten Inhalte stehen, außer dem Dateinamen, dem Anwendungstyp und dem Speicherort, ist es fast unmöglich Content in dieser Art und Weise zu handhaben. Auch Abfragen, wie bei einer Datenbank üblich, oder das Speichern von zusätzlichen Metadaten sind

nur schwer realisierbar. Das Dateisystem allein sollte daher lediglich für überschaubare Mengen von Dokumenten eingesetzt werden.

Im Gegensatz dazu ist der Einsatz von relationalen Datenbanken bei Entwicklern sehr beliebt, wenn es um das Ablegen von Content geht. Dies beruht auf der Tatsache, dass die meisten größeren Anwendungen bereits eine Datenbank verwenden und sich dadurch der Mehraufwand in Grenzen hält. Mit Hilfe von eigenen Datentypen den sogenannten Blobs⁴ ist es möglich durch Nutzung der gleichen Schnittstellen Dokumente abzulegen. Dabei sind mögliche Vorteile in der Verwendung datenbankspezifischer Funktionen wie Backup und Replikation zu sehen. Diese müssen jedoch den Nachteilen, welche sich durch mangelnde Skalierbarkeit und Performance-Einbußen, vor allem bei hohen Nutzeranzahlen, bemerkbar machen, gegenübergestellt werden.

Heutzutage verwendet deshalb nahezu jedes Content-Management-System eine Verbindung aus beiden Ansätzen, die sogenannte Referenz-Datenbank-Architektur, welche eine Referenz jedes Informationsobjektes und dazugehörige Metadaten in einer Datenbank verwaltet. Die Referenz erlaubt es dabei ein Dokument jeder Zeit in einem externen Speichersystem wiederzufinden und dem Anwender bereitzustellen. Hinzukommend werden Dienste bereitgestellt, die durch eine Kombination von Befehlen der Data Manipulation Language⁵ (SELECT, INSERT, UPDATE und DELETE) einen gezielten Datenzugriff erlauben und durch eine Verwendung innerhalb der Anwendungsschicht die Anforderungen an ECM erfüllen [2] S. 453-455. Beispiele für die Wirkungsbereiche solcher Dienste sind:

- Abfrage und Volltextsuche
- Auditing
- Berechtigungskontrolle
- Klassifikation und Tagging
- Hierarchisches Navigieren
- Metadatenextraktion
- Versionskontrolle

⁴ Akronym für „Binary Large Objects“.

⁵ Eine Datenbanksprache zur Manipulation der Tabellendaten.

3. Referenzlösung Alfresco

Mit Alfresco ist ab dem Jahr 2005 eine Open Source Alternative für Enterprise-Content-Management entstanden, welche andere kommerzielle Produkte in Eigenschaften wie Stabilität, Funktionalität, Anpassbarkeit und Reduzierung der Kosten übertreffen soll [3]. Um den Einsatz von Alfresco für die Verwaltung archivierter Dokumente zu begründen, sollen in diesem Abschnitt die Vorteile der Softwarelösung aufgezeigt werden.

3.1. *Total Cost of Ownership*

Das Ziel eines jeden Unternehmens ist das Streben nach Wirtschaftlichkeit. Um dies zu erreichen genügt es nicht nur den Gewinn zu maximieren, sondern auch bestehende Kosten zu minimieren oder durch das Treffen unternehmerischer Entscheidungen anfallende Ausgaben zu reduzieren. Die Anschaffung von ECM-Systemen gilt traditionell als kostenintensiv betreffend Nutzerlizenzen, Auslieferung und Wartung. Ebenfalls die Abhängigkeit von den Herstellern proprietärer Lösungen stellt, aufgrund komplexer Preismodelle und mangelnder Transparenz, seit jeher ein großes Problem dar. Mit dem verstärkten Gebrauch von Open Source Produkten im Bereich Enterprise-Content-Management, vorwiegend im Zusammenhang mit der Entwicklung des Web 2.0, konnte eine günstige Alternative für die Verwaltung von Content geschaffen werden. Alfresco versucht deshalb in einem White Paper ein Verständnis für die finanziellen Vorteile, welche der Einsatz von Open Source mit sich bringt, zu vermitteln. Unter dem Aspekt „do more with less“ gelingt es Alfresco sich von Konkurrenzprodukten wie Documentum, Microsoft SharePoint oder OpenText zu differenzieren und neue Wege einzuschlagen. Dabei liegt der Fokus hauptsächlich auf der Entscheidungsfreiheit des Kunden bei der Wiederverwendung existierender Hard- und Software, auf der Senkung von Integrationskosten durch die Unterstützung standardisierter Schnittstellen und auf einfache Preismodelle. Dies führt zu einer Minimierung der Total Cost of Ownership⁶ und ermöglicht somit durch den Einsatz dieses Produkts die Einsparung eines Großteils der Kosten im Vergleich zu anderen ECM-Lösungen [4].

⁶ Ein Verfahren zur Abschätzung der Kosten bei der Anschaffung von Investitionsgütern.

3.2. *Enterprise Application Integration (EAI)*

Um negativen Auswirkungen wie Abteilungs- und Funktionsgrenzen entgegenzuwirken, bietet das EAI-Konzept mittels verschiedener Methoden die Möglichkeit vorhandene Geschäftsanwendungen zu vernetzen. Der Integrationsgegenstand beschreibt dabei das vom Systemarchitekten angestrebte Ziel der Daten-, Anwendungs- oder Prozessintegration. In Bezug auf die Thematik Enterprise-Content-Management und den Zweck dieser Arbeit steht die Schaffung einer vollständigen sowie unternehmensweiten Datenbasis, d.h. das Ziel der Datenintegration, im Mittelpunkt. Dadurch werden Anwendungsfälle wie Archivierung, bei der abgelegte Dokumente zu einem späteren Zeitpunkt gesucht und wiedererlangt werden können, möglich. Möchte man dies ohne Veränderung der Systemlandschaft erreichen, so beschreibt EAI einen Ansatz, welcher eine Realisierung von Kommunikationskanälen zwischen den einzelnen Anwendungen vorsieht und einen Austausch von Informationen ermöglicht. Dies vermeidet wiederum neben der Redundanz von Daten, auch einen zusätzlichen Arbeitsaufwand bei der Pflege von Dokumenten bzw. Content und beugt einer Entstehung von Fehlerquellen vor. Mit einer Vielzahl von Schnittstellen kann Alfresco als eine zentrale Integrationsinstanz, die sogenannte „EAI-Middleware“, fungieren, so dass bei der Entwicklung angrenzender Anwendungen der Schwerpunkt auf Funktionalität anstatt auf Datenhaltung gelegt werden kann. [5] S. 1-3.

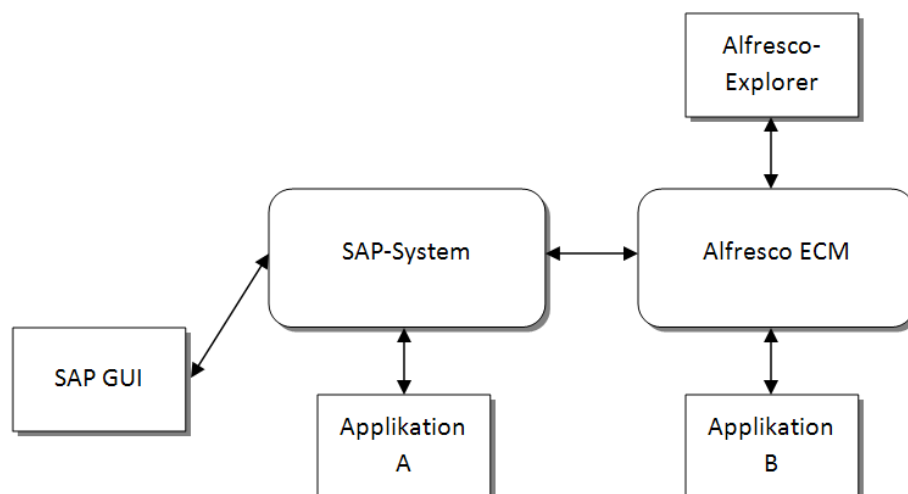


Abbildung 3: Integrationstopologie - Hub & Spoke am Beispiel der angestrebten Systemlandschaft

Durch den Einsatz dieser Topologie entsteht eine „Informationsdrehscheibe“, welche sowohl technische Vorteile als auch einen Nutzen für den Anwender mit sich bringt. Die technische Zweckdienlichkeit umfasst eine Reduzierung der Komplexität durch die verringerte Anzahl von Schnittstellen zwischen den Systemen, einen modularen Aufbau des Gesamtsystems sowie Flexibilität, Wiederverwendbarkeit und Nutzung von Standards. Auf Anwenderseite ergeben sich Vorteile in Form von geringeren Kosten durch verkürzte Bearbeitungszeiten sowie verbesserte Zusammenarbeit. Des Weiteren wird der einfache Austausch einzelner Softwarekomponenten garantiert, da die Kommunikation ausschließlich über die Middleware erfolgt. Betrachtet man all diese Merkmale, so lässt sich feststellen, dass die Entstehung eines unternehmensweiten Informationsflusses begünstigt wird [6].

3.3. *Einsatz von Frameworks*

Bei der Konzipierung fast jeden Softwaresystems steht die Lösung individueller Problemstellungen im Vordergrund. Um den Entwicklungsaufwand so gering wie möglich zu halten, versucht man einen großen Bereich des geplanten Systems durch den Einsatz von Frameworks⁷ abzudecken und auf Implementierungen häufig verwendeter Komponenten wie z.B. Persistenz, Logging oder XML-Verarbeitung zurückzugreifen. Dadurch kann sich vermehrt auf die Umsetzung der Geschäftslogik konzentriert werden. Alfresco lebt genau diese Philosophie. Mit Open Source Frameworks wie Spring, Hibernate, jBPM oder MyFaces gelingt es bekannte Strukturen und Konzepte zu schaffen, die es Java-Entwicklern erlaubt das System zu überblicken und mit standardisierten Arbeitsweisen gestellte Anforderungen zu realisieren.

3.4. *Systemarchitektur*

Wie bereits in Abschnitt 2.4 erläutert, soll die Architektur eines ECM-Systems nun anhand der Referenzlösung Alfresco belegt werden.

Die Systemarchitektur ist, wie man in der folgenden Abbildung sieht, klar in mehrere Schichten unterteilt. Auch hier stellt das Content-Repository den Kern der Anwendung dar und ermöglicht durch ein hohes Maß an Konfigurierbarkeit die

⁷ Eine Menge kooperierender Klassen, die ein wieder verwendbares Design für einen spezifischen Anwendungsbereich vorgeben [23] S. 4.

Anbindung verschiedener Datenbanken sowie die Adressierung unterschiedlicher Dateisysteme. Die sogenannten „Repository Foundation Services“ greifen auf die konkrete Implementierung zu und bilden somit die Schnittstelle zur Anwendungsschicht. In dieser finden sich neben den üblichen ECM Komponenten (Dokumentenmanagement, Kollaboration, Web-Content-Management, Records-Management und Workflow-Management) eine Vielzahl von Schnittstellen, welche eine Integration in vorhandene Softwareprodukte gewährleisten. Der Alfresco Explorer⁸ befindet sich oberhalb der Anwendungsebene und realisiert Präsentation sowie Steuerung der Systemarchitektur durch den Einsatz von HTML und der JSP⁹ Technologie.

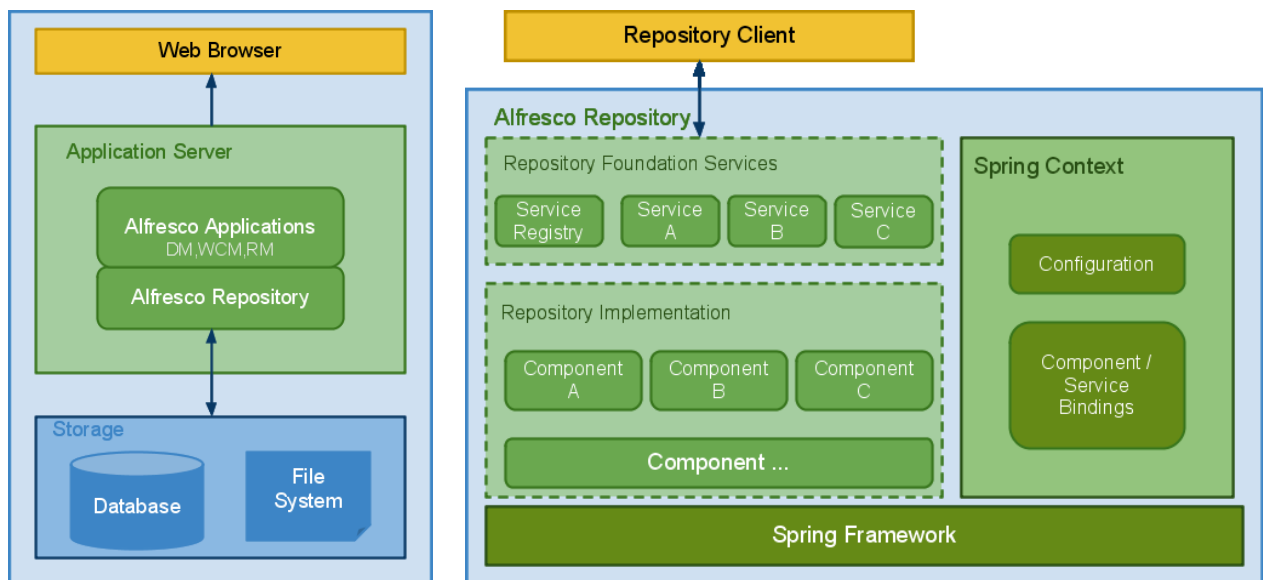


Abbildung 4: Alfresco Systemarchitektur, (Quelle: Alfresco Wiki, Alfresco Repository Architecture, 11.05.2009)

Durch die Verwendung des Spring Frameworks gelingt es nicht nur die Schichten an sich zu trennen, sondern auch innerhalb der einzelnen Ebenen Modularität zu gewährleisten. Eine Implementierung eines beliebigen vordefinierten Interfaces ermöglicht es Module ohne Auswirkungen auf existierende Bestandteile auszutauschen oder das System zu erweitern. Dieses Prinzip nennt man lokale Stetigkeit bei Änderungen. Innerhalb von Spring wird die angesprochene Kapselung

⁸ Der von Alfresco bereitgestellte Web-Client, welcher Dokumentenmanagement, Kollaboration, Benutzerverwaltung etc. unternehmensweit verfügbar macht.

⁹ Eine Programmiersprache zur dynamischen Erzeugung von HTML-Seiten [21].

mit Hilfe des Entwurfsmusters Dependency Injection¹⁰ erreicht, was zu einer erheblichen Reduktion der Softwarekomplexität und somit der Fehlerquellen führt.

3.5. Anpassbarkeit an spezielle Bedürfnisse

Aufgrund der Tatsache, dass die Anforderungen an Enterprise-Content-Management von der jeweiligen Unternehmung und dem geplanten Einsatzzweck abhängen, können diese durchaus sehr unterschiedlich sein. Alfresco bietet daher verschiedene Mechanismen, um das Softwareprodukt an spezielle Bedürfnisse anzupassen.

3.5.1. Der Open Source Weg

Überall dort, wo geschlossene Systeme existieren, werden die Vorteile von Open Source Lösungen deutlich. Entwickler erhalten die Chance Fehler selbst zu beheben, das System zu erweitern oder komplette Komponenten auszutauschen. Daraus resultiert nicht nur ein transparentes Softwaresystem, sondern auch eine Hersteller- bzw. Anbieterunabhängigkeit, welche von den meisten Unternehmen angestrebt wird.

Durch den Einsatz zahlreicher Open Source Frameworks bietet Alfresco auch die Möglichkeit die Implementierung dieser Rahmenwerke zu manipulieren und somit das Systemverhalten zu beeinflussen ohne eine Änderung des eigentlichen Quellcodes vorzunehmen.

3.5.2. Content-Modelling

Damit Inhalte effektiv verwaltet werden können, ist der Einsatz eines Modells sinnvoll, welches die Strukturierung sowie Abstraktion von Content durch die Definition von Klassen und die Beziehungen zwischen den jeweiligen Elementen beschreibt. Basierend auf diesen Metadaten ergeben sich nun systemweite Handlungsweisen betreffend Datenhaltung, Anwendungslogik und Präsentation für jede definierte Art von Content. Ob ein solches Modell und in welchen Ausmaßen es erstellt werden muss, ist vom Umfang eines Projektes abhängig. Alfresco bietet mit seinem Out-of-the-Box Repository Data Dictionary¹¹ lediglich eine Handvoll

¹⁰ Eine bewährte Lösung von Entwurfsproblemen in der Softwarearchitektur und -Entwicklung, bei der Abhängigkeiten zwischen Objekten minimiert werden.

¹¹ Ein Schema, welches herkömmliche Strukturen wie Ordner oder Dokumente definiert.

wiederverwendbarer Strukturen und somit eine Ausgangsbasis für die Content-Modellierung. Da sich XML für den Entwurf komplexer Datenstrukturen etabliert hat, bildet die vom W3C¹² herausgegebene Spezifikation auch in diesem Fall den technischen Rahmen bei der Definition des Modells.

3.5.3. Konfiguration des Systems

Die Bearbeitung der Dateien innerhalb des Classpaths¹³, welche beim Systemstart geladen werden, bietet ebenfalls eine Möglichkeit Änderungen an Alfresco vorzunehmen. So können nicht nur Datenbank- oder Repository-relevante Einstellungen vorgenommen werden, sondern auch solche betreffend der Benutzeroberflächen und des Systemverhaltens z.B. Vorgehensweisen beim Auftreten gewisser Dokumentenarten.

¹² Das World Wide Web Consortium ist eine internationale Gemeinschaft, welche durch die Veröffentlichung von Standards die stetige Entwicklung des Webs sicherstellt [22].

¹³ Eine Pfadangabe, in der eine Laufzeitumgebung nach benötigten Artefakten wie z.B. Bibliotheken oder Konfigurationsdateien sucht.

4. Dokumente im Unternehmen

In diesem Abschnitt soll die Bedeutung von Informationen und Dokumenten innerhalb von Unternehmen besprochen werden. Dabei wird auf die Notwendigkeit der Begleitung von Geschäftsprozessen durch Content-Management eingegangen und die Archivierung als ein Anwendungsfall diskutiert.

4.1. *Von der Vision zur Realität*

Seit einigen Jahren zählt die Information in den meisten Unternehmungen zu einem der entscheidendsten Wettbewerbsfaktoren. Dokumente oder Akten beinhalten einen Großteil der Informationen und spielen somit eine unumstrittene Rolle bei der Bewältigung des Tagesgeschäfts. Heutzutage liegen immer seltener Dokumente ausschließlich in Papierform vor, da die Digitalisierung bereits zu weiten Teilen abgeschlossen ist und neue Unterlagen oftmals direkt in eine elektronische Form überführt werden [7] S. 13. Deshalb stellen sich nun innerhalb vieler Organisationsstrukturen anschließende Fragen: Welche Dokumentenmanagement-Lösung ist für die bestehenden Anforderungen am geeignetsten und wie kann eine Integration in vorhandene betriebswirtschaftliche Anwendungen erfolgen?

4.2. *Bedeutung innerhalb von Geschäftsprozessen*

Zur Beantwortung der im vorherigen Punkt gestellten Fragen, ist es notwendig die Bedeutung von Dokumenten bei der Handhabung von Geschäftsprozessen zu erkennen. Als erstes sind dabei entscheidende Begrifflichkeiten, welche im direkten Zusammenhang mit der Thematik stehen, zu klären.

4.2.1. Was ist ein Geschäftsprozess?

Bei einem Geschäftsprozess handelt es sich um einen Vorfall, der einen definierten Anfang und ein definiertes Ende besitzt. Dieser ist immer auf ein bestimmtes Unternehmensziel ausgerichtet und stellt einen Teil der Wertschöpfungskette dar. Der Ablauf eines Geschäftsprozesses wird durch den Input von erforderlichen Daten sowie durch das Eintreten von Ereignissen beeinflusst. Während der Durchführung werden die zuvor modellierten Aktivitäten ausgeführt. Im Rahmen dessen liegt der Transport von Informationen im Fokus, da das Ergebnis jeder einzelnen Funktion den Eingang der folgenden Aktionen beschreibt [8] S. 7.

4.2.2. Was ist Workflow-Management?

Um zu verstehen, welche Aufgabenbereiche das Workflow-Management umfasst, ist es erforderlich den Begriff Workflow zu konkretisieren.

Ein Workflow stellt die technische Sicht eines Geschäftsprozesses dar und unterstützt diesen bei seinem Ablauf im Ganzen oder lediglich in ausgewählten Teilen. Eine Folge von vordefinierten Aktivitäten beschreibt den logischen Ablauf und bestimmt Merkmale wie Komplexität und Detaillierungsgrad eines Arbeitsflusses.

Das Workflow-Management beschäftigt sich neben der Verwaltung und der Durchführung der existierenden Workflows zunächst mit der Modellierung neuer Abläufe. Im Sinne der Analyse wird nach dem Entwurf üblicherweise eine Simulation der erstellten Workflows angesetzt, um Schwachstellen und Verbesserungsmöglichkeiten schon vor der eigentlichen Integration in die bestehenden Prozesse zu erkennen. Ist dies geschehen, so kann der Arbeitsfluss nun mit Hilfe einer Workflow-Engine¹⁴ genutzt werden [8] S. 10-13.

4.2.3. Geschäftsprozesse und Dokumente

Ein Grundgedanke, der hinter jeder Unternehmung steht, ist das ökonomische Prinzip [9] S. 37-39. Betrachtet man die Tatsache, dass die Gesamtheit und das Zusammenspiel der existierenden Prozesse das eigentliche Geschäft definieren, so müssen auch die einzelnen Handlungsweisen eine der drei folgenden Maximen anstreben:

- Das generelle *Extremumprinzip* beinhaltet die Realisierung eines möglichst günstigen Verhältnisses zwischen Aufwand und Ertrag.
- Beim *Maximumprinzip* soll mit einem gegebenen Aufwand ein möglichst hoher Ertrag erzielt werden.
- Der Gedanke des *Minimumprinzips* ist es, einem bestimmten Ertrag mit möglichst geringem Aufwand zu erzielen.

¹⁴ Ein Programm zum Ausführen von Workflows.

Um eines der genannten Ziele zu erreichen, stellt der Einsatz von Dokumenten als Informationsträger und -halter innerhalb von Workflows eine bereits bewährte Lösung dar [7] S. 22. Im Folgenden sollen anhand von zwei Beispielen die unterschiedlichen Einsatzmöglichkeiten verdeutlicht werden, die es sinnvoll machen Geschäftsprozesse und Dokumentenmanagement zu verbinden.

Das erste Beispiel zeigt den stark vereinfachten Managementprozess der Urlaubsbeantragung in Form eines Flussdiagrammes. Das Dokument *Urlaubsantragsformular* dient dabei als Auslöser des Vorgangs und stellt den Input für die erste Aktivität dar. Das Formular beinhaltet die Informationen des Antragstellers, welche nach dem Eingang des Dokuments geprüft werden sollen. Eine Durchführung dieses Prozesses ohne die notwendigen Informationen wie den Zeitraum oder die Angaben zum Arbeitnehmer wäre unvorstellbar.

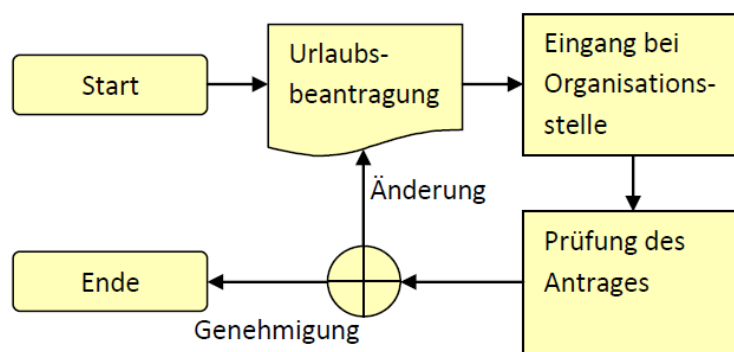


Abbildung 5: Flussdiagramm: Urlaubsbeantragung

Im zweiten Beispiel sieht man einen Kernprozess eines jeden Unternehmens, die Kundenanfrage. Anhand des folgenden Flussdiagrammes ist zu sehen, dass das Dokument *Bestellung* als Ergebnis des Prozesses betrachtet werden kann, sofern man den sich anschließenden Geschäftsprozess vernachlässigt. Ist dies nicht der Fall, so bekommt das Dokument eine völlig andere Bedeutung. Es ist nun Auslöser des nachfolgenden Bestellprozesses [7] S. 23.

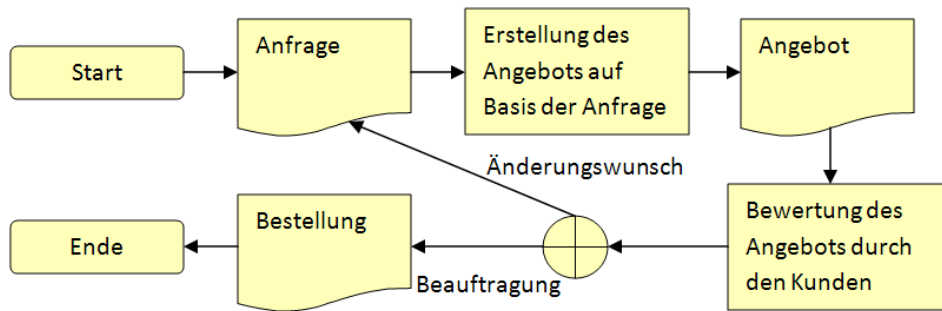


Abbildung 6: Flussdiagramm: Anfrage eines Kunden

Beide Beispiele verdeutlichen die Bedeutung von Dokumenten im Unternehmen sowie das Einhergehen bei der Durchführung von Geschäftsprozessen. Die Art und Weise, wie ein Dokument innerhalb eines Workflows verwendet wird, ist durch die Analyse der einzelnen Prozesse zu ermitteln, und kann somit von Fall zu Fall stark variieren.

4.2.4. Rechtliche Aspekte

Sobald Schriftstücke den Unternehmenserfolg aufzeigen, als Beleg für Geschäftsvorfälle dienen oder personenbezogene Informationen enthalten, ist die Einhaltung des rechtlichen Rahmens zu berücksichtigen. Die folgende Zusammenfassung zeigt die wesentlichen Aspekte, welche auf Grundlage von Gesetzestexten, Verordnungen und Richtlinien einzuhalten sind [7] S. 18-20.

- *Ordnungsmäßigkeit* ist bei der Erstellung, Verwaltung und Archivierung jedes relevanten Dokumentes sicherzustellen. Sie beinhaltet die Nachvollziehbarkeit aller Aktionen, die ein Dokument und dessen Inhalt betreffen.
- *Vollständigkeit* soll garantieren, dass Dokumente in ihrer Gesamtheit vorliegen.
- *Datensicherheit* beschreibt Sicherungsmaßnahmen, welche die Daten vor unberechtigten Zugriff, vor Entwendung oder Zerstörung und vor Außeneinwirkungen schützen sollen.
- *Datenschutz* befasst sich im Rahmen des Bundesdatenschutzgesetzes mit den einzuhaltenden Rechten bei der Handhabung von personenbezogenen Daten.

- *Aufbewahrungsfristen* sind rechtlich vorgeschriebene Fristen, die ein Schriftstück je nach Kontext entweder mindestens aufgehoben werden muss oder höchstens aufgehoben werden darf.
- *Gesetzlicher Zugriff* stellt sicher, dass auskunftsberechtigte Behörden oder Instanzen einen Zugriff auf betreffende Dokumente erhalten.
- *Beweiskraft vor Gericht* ist nur gewährleistet, wenn ein elektronisches Dokument als Beweismittel anerkannt wird. Da es nicht den gleichen Status wie ein Dokument in Papierform besitzt, sind einige Voraussetzungen zu erfüllen. So ist es erforderlich, dass das reproduzierte Dokument mit dem Original übereinstimmt, der Archivierungsvorgang keine Manipulationen zulässt und vollständig dokumentiert ist.
- *Verfahrensdokumentation* beinhaltet die lückenlose Dokumentation wie mit den elektronischen Dokumenten umgegangen werden soll sowie die Protokollierung der ordnungsgemäßen Anwendung des Verfahrens. Sie gilt somit als Grundlage der zuvor genannten Anforderungen, Verordnungen und Richtlinien.

4.3. *Archivierung als Anwendungsfall*

Noch immer denken viele IT-Verantwortliche bei dem Wort „Archiv“ an staubige Räume voller Aktenordner oder Datenträger, was sich oftmals auch mit dem geringen Stellenwert von Archivierung im Unternehmen verbinden lässt. Ein gut geführtes Archiv unterstützt nicht nur effizient Geschäftsprozesse, sondern bringt noch weitere Vorteile mit sich, die in diesem Zusammenhang behandelt werden sollen.

4.3.1. Anforderungen

Die Anforderungen an ein elektronisches Archivsystem gehen weit über die rechtlichen Aspekte hinaus und werden durch technische sowie funktionale Merkmale definiert. So sollten bei der Langzeitspeicherung von Dokumenten und Informationen die Ansätze der Migration, um auch nach Jahren den Zugriff auf Daten zu ermöglichen, verfolgt werden, anstatt auf die Emulation früherer Software- und Hardwareumgebungen zu setzen oder solche Soft- und Hardware vorzubehalten.

Des Weiteren ist aus technischer Sicht vor allem auf die Wahl geeigneter Datenformate zu achten. Hierbei spielen Kriterien wie Lesbarkeit, Wiedergabefähigkeit, Verbreitungsgrad oder Reife der Formate eine wichtige Rolle. Im Gegensatz zu den technischen befassen sich die funktionalen Anforderungen in erster Linie mit standardisierten Schnittstellen für eine Integration der elektronischen Archive in beliebige Anwendungen sowie mit Konzepten für die Zugriffssicherung und die Bereitstellung von Informationen. Damit sollen letztendlich die Bedürfnisse der Anwender, welche sich bei der Bewältigung des Tagesgeschäftes erkennen lassen, erfüllt werden [10].

4.3.2. Einsatzziele

Wie in diesem Kapitel bereits besprochen, ist die Bedeutung von Dokumenten als Informationsträger begleitend zum oder als Ergebnis eines Geschäftsprozess nicht abzustreiten. Bereits beim Entwurf von Arbeitsabläufen sollten sich daher Gedanken gemacht werden, was mit den Dokumenten am Ende eines jeden Workflows geschehen soll und ob überhaupt eine Notwendigkeit besteht diese aufzubewahren. Eine Möglichkeit stellt natürlich die langfristige oder dauerhafte Sicherung digitaler Unterlagen dar. Die Ziele dieses Vorhabens beinhalten sowohl die Einhaltung der Aufbewahrungsfristen und den Schutz der Daten vor unerlaubten Zugriffen, als auch die Stärkung und Erweiterung des Produktivsystems durch Integration des Archivs. Dies macht sich einerseits durch eine Optimierung von Geschäftsprozessen bemerkbar, so dass Mitarbeiter Dokumente direkt einsehen, herunterladen oder ausdrucken können, ohne die Unternehmenslösung zu verlassen. Andererseits steigert sich die Systemleistung aufgrund der Tatsache, dass ausschließlich Referenzen auf das Dokument in der Anwendung gehalten werden und die eigentlichen Daten auf ein oder mehrere externe Speichermedien verteilt sind [11].

4.3.3. Vorteile gegenüber Produktivsystemen ohne Archiv

Zusammenfassend kann eine Vielzahl von Vorteilen aufgezählt werden, welche mit dem Einsatz von Archivsystemen in einem direkten Zusammenhang stehen.

- Beschleunigter Datenbankbetrieb und damit hohe Systemverfügbarkeit
- Ressourcengewinn durch verringerten Bedarf an Speicherplatz, Arbeitsspeicher und Prozessorleistung
- Zusammenführung von Informationen

- Unterstützung der Geschäftsprozesse im Unternehmen
- Vermeidung von Medienbrüchen bei der Verarbeitung von Informationen
- Effizientes Arbeiten ohne lange Dokumentensuche

4.4. *ECM eine vielseitige Lösung*

Bei der Beantwortung der anfangs gestellten Fragen kann Enterprise-Content-Management als eine mögliche Antwort bei der Verbindung von Geschäftsprozessen und Dokumentenmanagement sowie bei der Integration von Anwendungen sein. Hierbei stellen die in Abschnitt 2.3 genannten Bestandteile der Verwaltungs-Komponente das Grundgerüst für ein erfolgreiches Zusammenspiel dar, da diese einen Großteil der Anforderungen erfüllen. Neben dem klassischen Verwalten ist es mit dem verfügbaren Workflow-Management möglich den existierenden Dokumenten Arbeitsabläufe zuzuweisen. Erfolgt die Zuweisung automatisch sind viele unterschiedliche Szenarien denkbar, die einerseits den Zeitpunkt der Zuordnung und andererseits den Workflow spezifizieren. So könnte das Starten eines Prozesses z.B. beim Erstellen eines Dokuments im Content-Repository oder beim Auftreten in einem bestimmten Ordner erfolgen und der einzusetzende Workflow anhand der Dokumentenart identifiziert werden. Aufgrund der Tatsache, dass an einem Geschäftsprozess in den meisten Fällen mehrere Organisationsstellen beteiligt sind, bietet ECM auch hier in Form der Nutzerverwaltung die passende Komponente.

Endet der Arbeitsablauf einer angrenzenden Unternehmenslösung damit, dass ein Dokument im Enterprise-Content-Management-System abgelegt wird, so kann dieses als Informationsdienst genutzt werden, indem es die passenden Schnittstellen als ein Service-Provider bereitstellt. Besonders bei hohen Anforderungen an die Verteilung von Content betreffend unterschiedlicher Endanwender und multipler Kommunikationskanäle kann ein Produkt wie Alfresco enorme Vorteile mit sich bringen. So könnte beispielsweise ein und dieselbe Datenbasis für ein Enterprise-Portal, innerhalb von Desktop-Anwendungen oder auf mobilen Endgeräten genutzt werden.

5. Dokumentenmanagement mit SAP

Durch die Entstehung verschiedener Technologien ist es seit einigen Jahren möglich innerhalb von SAP die Funktionen eines DMS abzubilden. Aufgrund der Tatsache, dass in fast jedem Geschäftsprozess Dokumente eine entscheidende Rolle spielen, haben sich diese von SAP mitgelieferten Softwarekomponenten in vielen Unternehmen etabliert. In diesem Kapitel sollen die grundlegenden SAP-Technologien behandelt und Möglichkeiten der Integration von Drittsystemen aufgezeigt werden.

5.1. Überblick

Damit effizientes Dokumentenmanagement möglich wird, sind in jedem SAP-System standardisierte Funktionalitäten vorhanden. Die Bausteine des klassischen DMS wie Eingang/Klassifikation, Ablage/Archivierung oder Recherche werden in SAP durch Dienste für Schriftverkehr, Anwendungsintegration und natürlich Workflow ergänzt. Die folgende Abbildung soll eine Vorschau zur Thematik geben:

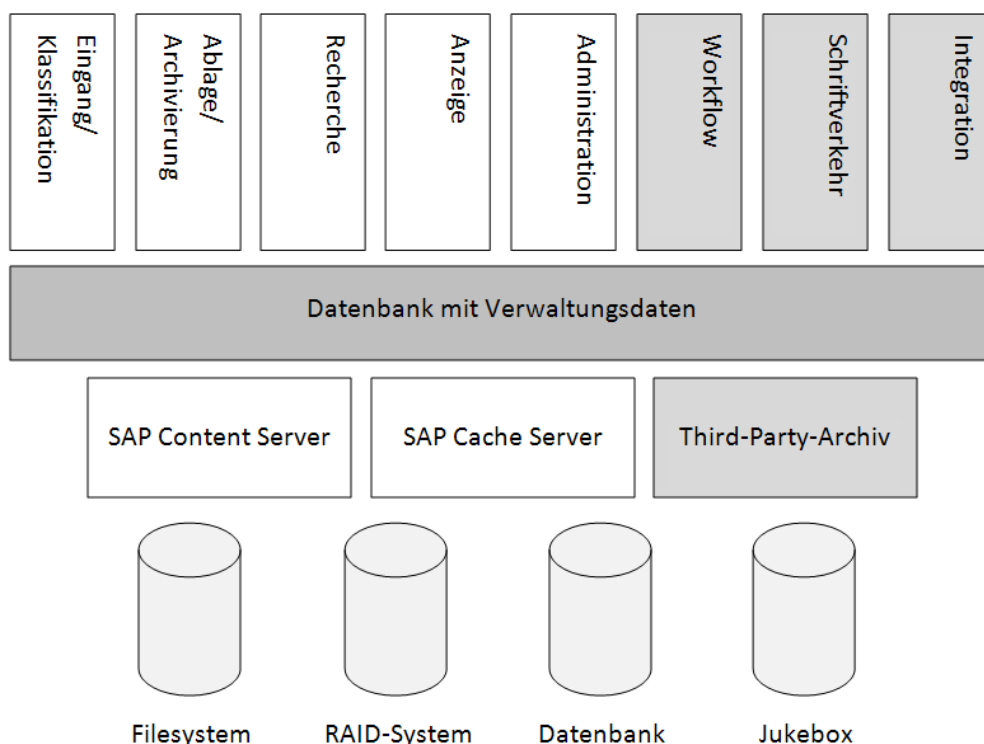


Abbildung 7: Überblick Dokumentenmanagement mit SAP (Quelle: Heck, Rinaldo, Geschäftsprozessorientiertes DMS mit SAP, 2009 S. 33)

5.2. Plattform „NetWeaver“

Um den Überblick bei der Vielzahl von SAP-Anwendungen zu bewahren, erscheint es sinnvoll mit der Basis zu beginnen, welche die Plattform für jegliche Geschäftsanwendung bildet. Mit SAP NetWeaver bietet die SAP AG eine technologische Gesamtbasis, welche die Grundlage für unternehmensrelevante Softwarekomponenten wie z.B. Business Intelligence oder Enterprise Portal darstellt [7] S. 29-32. Eine Aufteilung des Produktes kann dabei in vier Schlüsselbereiche erfolgen:

- Die *Applikationsplattform*, realisiert durch den NetWeaver Application Server, gilt als Fundament von SAP NetWeaver und lässt sich wiederum in vier wichtige Funktionsbereiche untergliedern.

Tabelle 2: Funktionsbereiche des NetWeaver Application Servers

Funktion	Beschreibung
Plattformweite Services	Eine Reihe von plattformweiten Diensten ermöglicht den Austausch von Anwendungsdaten zwischen SAP-Systemen untereinander und mit Fremdsystemen sowie die Anbindung an verschiedene Datenbanken und Speichersysteme.
ABAP	ABAP dient zur Realisierung von Eigenentwicklungen sowie zur Modifikation von Standardanwendungen in SAP. Der Applikationsserver beinhaltet eine vollständige Entwicklungs- und Laufzeitumgebung für ABAP-basierte Anwendungen.
Java	Einen weiteren Bestandteil stellt die J2EE Plattform dar. Diese bietet einen komponentenbasierten Ansatz für den Entwurf, die Entwicklung und den Einsatz von Java Enterprise Applikationen im SAP Umfeld [12].
Business Services	Mit den im Applikationsserver integrierten anwendungsorientierten Diensten wird z.B. Dokumenten-, Workflow- oder auch Organisationsmanagement ermöglicht.

Die Business Services stellen die Technologien für die Verwendung von DMS-Funktionalitäten in SAP zur Verfügung und werden deshalb im Abschnitt **5.3** auszugsweise behandelt.

- Die *Integration von Prozessen* stellt einen weiteren Teil der NetWeaver Plattform dar. Spezielle Anwendungen ermöglichen neben der

Automatisierung von Geschäftsabläufen eine ganzheitliche Betrachtung und Optimierung aller Prozesse von Wareneingang, über die Produktion bis hin zum Versand.

- Aufgrund der Erfassung und Speicherung von immer mehr Daten werden im Rahmen der *Integration von Informationen* Komponenten zur Analyse großer strukturierter sowie zur Organisation unstrukturierter Inhalte bereitgestellt. Das Ziel dieser Werkzeuge ist es die, durch den verschärften Wettbewerb und die schwierigen wirtschaftlichen Bedingungen, immer komplexer werdenden Entscheidungsprozesse so gut wie möglich zu unterstützen.
- Einen wichtigen Punkt beschreibt die *Integration von Menschen*, was neben der Bereitstellung von umfangreichen Daten mit Hilfe von Portalen auch die Funktionalitäten zur synchronen und asynchronen Kommunikation, wie Instant Messaging, E-Mail oder Online-Foren, innerhalb von Teamstrukturen umfasst.

5.3. *Relevante Business Services*

Die SAP AG bietet, wenn es um das Thema Dokumentenmanagement geht, bereits ein komplettes Produkt-Portfolio, das Basistechnologien für jeden Prozessschritt beinhaltet. Da der SAP-Alfresco-Connector einen Einsatz von Alfresco als Archivlösung ermöglichen soll, ist lediglich ein Teil der Prozessbausteine für die Realisierung von Relevanz. Eine Abbildung soll die für diese Arbeit entscheidenden Komponenten hervorheben.

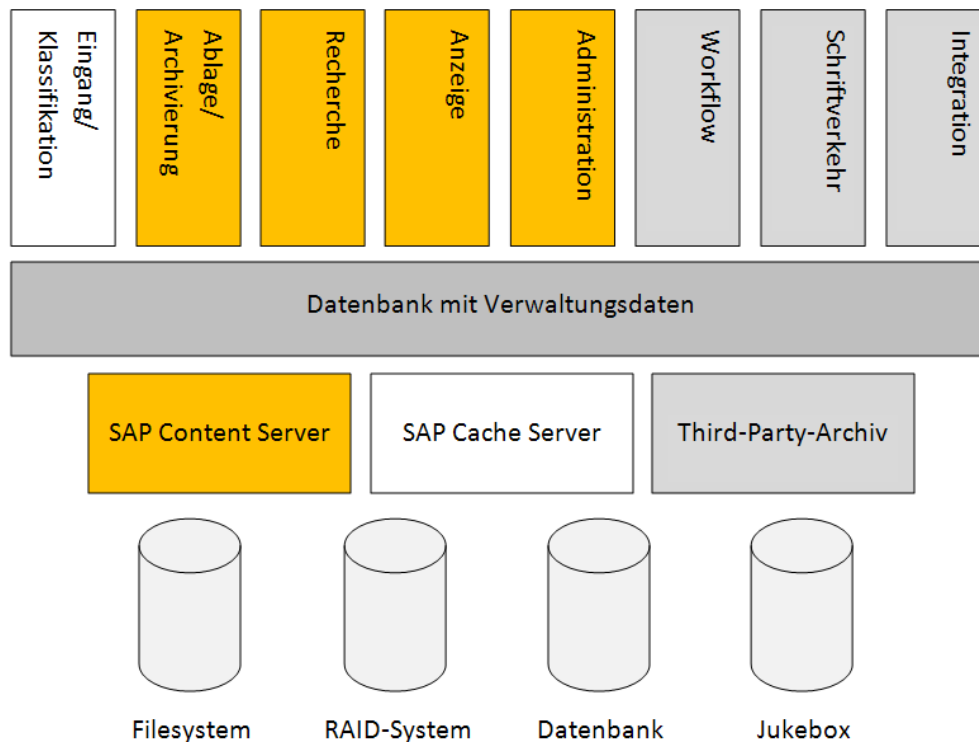


Abbildung 8: Relevante DMS-Komponenten

Um die Archivierung von Ausgangsbelegen oder die Verknüpfung von anderen Dokumenten mit Geschäftsobjekten zu ermöglichen, sind die orange gekennzeichneten Bausteine und die dahinter stehenden Dienste von Bedeutung.

5.4. **SAP ArchiveLink**

ArchiveLink beschreibt einen Service zur Verknüpfung von archivierten Dokumenten mit im SAP-System vorhandenen Anwendungsobjekten. Durch die technische Verknüpfung ist zu jeder Zeit ein einfacher Zugriff möglich, so dass mit Hilfe der Dienste eines Geschäftsobjekts die dazugehörigen abgelegten Dokumente direkt angezeigt werden können. Hierdurch werden zeitaufwendige Arbeitsabläufe wie das Suchen nach vorgangsbezogenen Informationen überflüssig und der Anwender kann sich voll und ganz auf seine eigentlichen Aufgaben konzentrieren. Die Kommunikationsschnittstelle SAP ArchiveLink stellt eine Basiskomponente dar, welche eine Anbindung von externen Komponenten wie z.B. Ablagesysteme durch die Nutzung der technischen Infrastruktur eines Content Servers (siehe Abschnitt 5.5) realisierbar macht [13] S. 11.

5.4.1. Verwaltungskonzept

Um eine Verbindung von abgelegten Dokumenten und Business-Objekten herzustellen, ist die Durchführung eines Grund-Customizings innerhalb von SAP notwendig [7] S. 39-40. Die nachfolgenden Schritte sind zwingend erforderlich:

➤ **Content-Repository pflegen (Transaktion OAC0)**

Bei einem solchen Repository handelt es sich um einen Ablageort zur Speicherung, Beschreibung und Verwaltung von Informationen jeglicher Art. Die Verwendung mehrerer Content-Repositories kann eine fachliche oder physikalische Trennung eines Archivs ermöglichen, welche oftmals Vorteile bei der Bewältigung großer Datenmengen mit sich bringt.

➤ **Dokumententypen pflegen (Transaktion OAD2)**

Das technische Format einer Datei (PDF, DOC, TXT, etc.) wird in SAP durch Dokumenttypen repräsentiert. Diese sind eine Voraussetzung für das Pflegen von Dokumentenarten.

➤ **Dokumentenarten pflegen (Transaktion OAC2)**

Da für verschiedene Dokumente unterschiedliche Verfahrensweisen betreffend des Ablagesystems, des Speicherortes und der Aufbewahrungsfristen existieren, wurden Dokumentenarten zur Untergliederung von Dokumenten eingeführt.

➤ **Verknüpfungen pflegen (Transaktion OAC3)**

Um die SAP ArchiveLink Schnittstelle letztendlich nutzen zu können, muss eine Zuordnung der zuvor gepflegten Objekte erfolgen. Mit Hilfe der Transaktion „Verknüpfung pflegen“ ist es möglich ein Objekttyp, der entsprechenden Dokumentart, einem Dokumenttyp, einem Content-Repository und einer Zieltabelle für die Verknüpfung zuzuweisen.

Eine Zuordnung von Geschäftsobjekten zu Archivdokumenten erfolgt durch das Schreiben von Datensätzen in der Verknüpfungstabelle während des laufenden Betriebs. Diese setzen sich aus Feldern für die Objektidentifikation des Business-Objekts, für die Dokumentidentifikation des dazugehörigen Dokuments sowie für verwaltende Informationen zusammen.

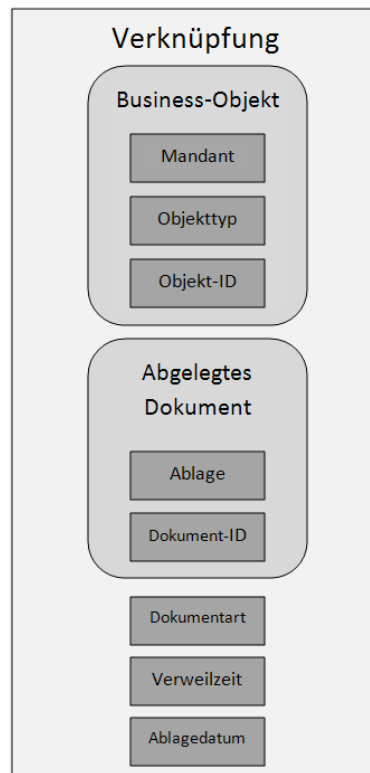


Abbildung 9: Verknüpfungsdaten (Quelle: SAP AG, SAP ArchiveLink, 2001 S. 54)

5.4.2. Ablagearten

Betrachtet man das angestrebte Fallbeispiel „Erfassung von ausgehenden Rechnungsbelegen“, erkennt man, dass es sich bei diesem Prozessablauf nicht um eines der zahlreichen ArchivLink-Szenarien für Eingangsdokumente handelt. Ausgehende Dokumente (Rechnungen, Mahnungen, Lieferscheine, etc.) definieren sich durch eine Erzeugung innerhalb von SAP-Anwendungskomponenten. Sie werden normalerweise ausgedruckt und per Post versandt. Dabei existieren die folgenden drei Ablagearten: Drucken (nur auf dem Drucker), Ablegen (nur in ein externes Ablagesystem) sowie Drucken und Ablegen. Die letzteren Beiden sind durch den Einsatz der ArchiveLink-Schnittstelle realisierbar [13] S. 87.

5.5. *SAP HTTP Content Server 4.5 Schnittstelle*

Der Content Server der SAP AG beschreibt ein Ablagesystem für dokumenten-zentrierte Anwendungen und Geschäftsprozesse. Mit Hilfe von allgemeinen Industriestandards wie HTTP oder BAPIs¹⁵ wird das Ziel verfolgt eine Schnittstelle für

¹⁵ Standardisierte Programmierschnittstellen von SAP Geschäftsobjekten.

die Kommunikation mit externen Systemen zu schaffen. Dabei ermöglicht eine Implementierung des SAP HTTP Content Server 4.5 Interfaces eine Integration von Drittsoftware in die Systemlandschaft. Das tatsächlich verwendete Speichermedium z.B. ein spezielles Enterprise-Content-Management-System bleibt dem Server bzw. dessen Realisierung überlassen. Es ist somit verborgen und kann, falls neue Anforderungen entstehen, jeder Zeit ausgetauscht werden.

5.5.1. Content-Modell

Um die Arbeitsweise eines Content Servers besser verstehen zu können, ist es angebracht einen Blick auf das Content-Modell zu werfen [13] S. 238. Dieses beabsichtigt eine eindeutige Verbindung der abgelegten Dokumente mit den Geschäftsobjekten. Bei der Identifikation von Zuordnungen sind folgende drei Begrifflichkeiten von besonderer Bedeutung:

- Ein Content-Repository dient als ein logischer Ablageort von Dokumenten und wird von der Serverkomponente anhand einer zugewiesenen ID erkannt. Da eine Zuordnung innerhalb von SAP erfolgt, handelt es sich um eindeutige Schlüssel, die es erlauben mehrere Repositorien durch einem Content Server verwalten zulassen.
- Der sogenannte Dokumentenkopf fasst mehrere Komponenten in sich zusammen und wird wiederum mit Hilfe der Dokument-ID eindeutig identifiziert.
- Eine Komponente repräsentiert eine Content-Einheit. Sie ist Bestandteil genau eines Dokumentenkopfes und besitzt eine eigene ID.

Das folgende Modell soll die Beziehungen zwischen einzelnen Objekten noch einmal verdeutlichen und den Aufbau eines Dokuments in SAP zeigen.

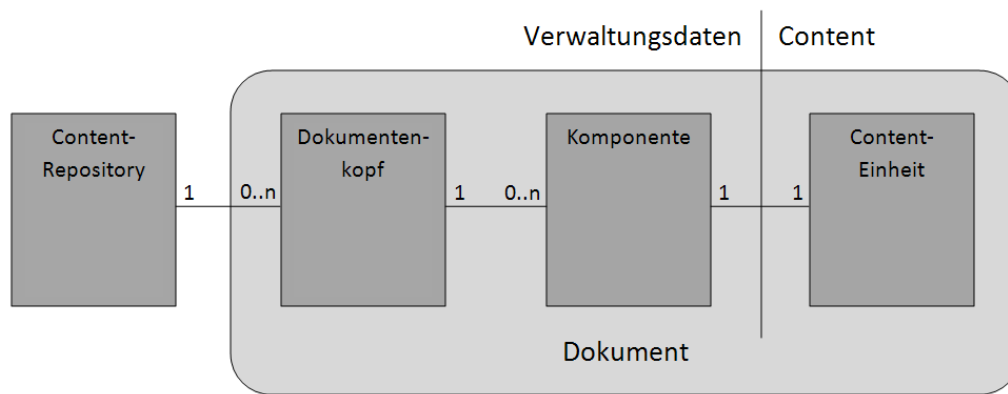


Abbildung 10: Dokumentenbegriff in SAP (Quelle: SAP AG, SAP ArchiveLink, 2001 S. 239)

Die Vorteile eines solchen Modells liegen in dessen vielseitigen Verwendung und der Möglichkeit nahezu jedes digitale Dokument abbilden zu können. Ein Beispiel dafür ist eine elektronische Bewerbung per E-Mail. Hier repräsentiert die Mail einschließlich der gesamten Metadaten wie Absender, Betreff oder Sendedatum den Dokumentenkopf. Die einzelnen Anhänge z.B. Bewerbungsschreiben, Lebenslauf oder Zeugnisse stellen die Komponenten des Dokumentes Bewerbung dar und die Inhalte dieser den eigentlichen Content [7] S. 35-36.

5.5.2. Arbeitsweise

Wie bereits erwähnt, findet die Kommunikation mit einem Content Server über das HTTP Protokoll statt. Die verschiedenen Content-Repositories und die darin aufbewahrten Dokumente werden durch URLs mit einer speziellen Syntax, die ebenfalls die auszuführenden Funktionen sowie die notwendigen Parameter beinhaltet, adressiert.

5.5.3. Funktionen

Die SAP Content Server HTTP Schnittstelle stellt neben den administrativen Funktionen hauptsächlich Funktionalitäten bereit, die auf den existierenden Dokumentenbestand zugreifen. Neben dem unterschiedlichen Auftreten von vorgeschriebenen und optionalen Parametern, kann eine Differenzierung vor allem anhand der Zugriffsarten erfolgen, wie die folgende Tabelle zeigen soll.

Tabelle 3: Content Server: Funktionsüberblick

Kommando	Wirkung	Zugriffsart
info	Abrufen von Informationen über das Dokument.	lesend

get	Holen (eines Bereiches) einer Content-Einheit einer Komponente.	lesend
docGet	Holen des gesamten Inhalts eines Dokumentes.	lesend
create	Neuanlegen eines Dokumentes.	erzeugend
mCreate	Neuanlegen mehrerer Dokumente.	erzeugend
append	Anhängen von Daten an eine Content-Einheit.	erneuernd
update	Modifizieren eines bestehenden Dokumentes.	erneuernd
delete	Löschen eines Dokumentes bzw. einer Komponente.	löschend
search	Such eines Textmusters in einer Content-Einheit.	lesend
attrSearch	Suche eines oder mehrerer Attribute innerhalb eines Dokumentes. (Suche innerhalb von Drucklisten)	lesend
putCert	Zertifikat des Clients (z.B. R/3-System) übergeben.	-
serverInfo	Abrufen von Informationen zum Content Server und den entsprechenden Content-Repositories.	-

Um die SAP ArchiveLink Schnittstelle in all ihren Facetten nutzen zu können, ist eine Implementierung der Funktionen in den verschiedenen Ausprägungen, d.h. unter Betrachtung der möglichen Parameterkombinationen, notwendig.

Nach einigen Tests mit Hilfe eines TCP¹⁶ Monitoring Werkzeuges konnte jedoch festgestellt werden, dass für die prototypische Realisierung des Fallbeispiels nur ein Teil der Funktionen vollständig zu implementieren sind. Damit der zeitlich begrenzte Rahmen eingehalten und eine Erfüllung weiterer Produktkriterien angestrebt werden kann, ist es in diesem ersten Entwicklungsschritt lediglich erforderlich die notwendigen Funktionalitäten umzusetzen.

¹⁶ Transmission Control Protocol (engl. für „Übertragungssteuerungsprotokoll“)

6. SAP-Alfresco-Connector

Dieses Kapitel befasst sich mit der Entwicklung des Connectors und den einzelnen Phasen des Entstehungsprozess.

6.1. Analyse

Innerhalb dieser Entwicklungsphase soll eine Abstraktion des realen Problems in Form eines Analysemodells erstellt werden. Ziel ist es das zukünftige System zu definieren und eine Abgrenzung des Softwareprojekts vorzunehmen.

6.1.1. Ziel / Produktzweck

Um die in der Zielsetzung (siehe Abschnitt 1.2) bereits erwähnten Anforderungen an das spätere Produkt erfüllen zu können, sind eine Reihe von Kriterien von Bedeutung. Die folgende Tabelle zeigt eine Unterteilung dieser je nach Relevanz in Muss-, Soll- und Kann-Kriterien. Da es sich bei einem Projekt um ein zeitlich begrenztes Vorhaben handelt, liegt der Fokus vorerst auf der Erfüllung der Muss-Kriterien.

Tabelle 4: Kriterien des SAP-Alfresco Connectors

Kriterium	Beschreibung
Muss	
	Der Connector muss als eine Webanwendung vorliegen, um in einem Java Servlet Container z.B. Apache Tomcat läuftig zu sein.
	Eine Implementierung der SAP Content Server HTTP Schnittstelle muss erfolgen.
	Die Kommunikation zwischen dem Connector und den verschiedenen Content-Repositories muss mit Hilfe des <i>Content-Management-Interoperability-Services (CMIS)</i> Standards erfolgen.
Soll	
	Die Administration der Software soll auf Basis der <i>Java Management Extensions (JMX)</i> Spezifikation erfolgen.
	Eine Basiskonfiguration des Connectors soll durch das Bearbeiten von *.properties-Dateien möglich sein.
Kann	

	Zusätzlich zu den abgelegten Dokumenten können Metadaten aus SAP-internen Tabellen durch den Einsatz des <i>SAP Java Connectors (JCO)</i> erlangt und gespeichert werden.
--	---

6.1.2. Produkteinsatz

Der Anwendungsbereich des Produktes befindet sich im Umfeld von SAP- und CMIS-konformen Content-Management-Systemen. Die Verwendung des Software-systems ermöglicht die Verwirklichung von Archivierungsaufgaben zwischen den genannten Systemen.

Die Zielgruppen für die entwickelte Lösung sind Unternehmen, welche SAP-Systeme im Einsatz haben und eine Realisierung von Archivierungsszenarien mit spezifischen CMS oder ECM-Systemen anstreben.

Das Einsatzgebiet der Software ist die Systemumgebung des jeweiligen Kunden. Ein Java Webapplication Server stellt dabei die Laufzeitumgebung dar. Das Softwaresystem kann ohne Aufsicht eines Bedieners verwendet werden und richtet sich nach den Betriebszeiten des Unternehmens.

6.1.3. Identifikation der Schnittstellen

Bei der Planung des Softwaresystems war es als zu Beginn notwendig Schnittstellen zu bestimmen, welche eine flexible Anbindung an die vorhandenen Systeme ermöglichen. Um dies zu gewährleisten ist oftmals der Einsatz von standardisierten Schnittstellen sehr sinnvoll, da diese von ähnlichen Lösungen unterstützt werden und somit eine Austauschbarkeit der angrenzenden Produkte erlauben. Des Weiteren existieren ausführliche Dokumentationen zu den jeweiligen Spezifikationen, welche bei Änderungen z.B. durch Release-Wechsel ebenfalls aktualisiert werden. Dies erleichtert es dem Entwickler die Software an die gegebene Schnittstellenversion anzupassen.

Betrachtet man den Produktzweck des SAP-Alfresco-Connectors, so fällt die Auswahl einer Schnittstelle auf Seiten des SAP Enterprise Resource Planing Systems nicht schwer. Um Ausgangsbelege von Geschäftsprozessen zu archivieren

oder externe Dokumente¹⁷ mit Business-Objekten zu verknüpfen, sieht die SAP AG die ArchiveLink-Schnittstelle (vgl. 5.4) vor. Diese lässt sich wie bereits erläutert, durch eine Implementierung des SAP Content Server HTTP Interfaces realisieren. Zu Seiten der Enterprise-Content-Management Lösung gestaltet sich die Verwendung einer spezifischen Schnittstelle, aufgrund der zahlreichen Anbindungsmöglichkeiten, welche Alfresco bietet, als nicht ganz so eindeutig. Im folgenden Abschnitt sollen die einzelnen Varianten vorgestellt und bewertet werden.

Beim Entwickeln gegen das Content-Repository von Alfresco gibt es eine Vielzahl von Vorgehensweisen, die es erlauben die bereitgestellten Dienste zu nutzen und einen Zugriff auf die gewünschten Informationen zu erhalten. Möglichkeiten dafür bieten die verschiedenen Remote-APIs¹⁸.

- Die Alfresco RESTful API stellt eine zustandslose HTTP-basierte Schnittstelle dar, welche es ermöglicht das Repository von Alfresco mit Hilfe von URIs¹⁹ als einen Informationsdienst zu verwenden [14]. Vorteilhaft ist hierbei die Tatsache, dass das HTTP selbst einen Kommunikationsstandard beschreibt und somit eine Implementierung portierbar macht. Auch die Verständigung zwischen Client und Server gestaltet sich als durchaus effizient. Betrachtet man die einfache Methodik, welche es erlaubt durch den gezielten Einsatz der verschiedenen HTTP-Kommandos (GET, PUT, DELETE, etc.) und parametrisierten Pfadangaben serverseitige Ressourcen zu manipulieren.
- Die Web Services API beschreibt eine weitere Möglichkeit, einen Zugang zum Content-Repository von unterschiedlichen Client-Umgebungen aus z.B. PHP, .NET, Java zu erlangen [15]. Ähnlich wie bei der Alfresco RESTful API spielt auch hier die Hard- und Softwareumgebung nur eine untergeordnete Rolle. Jedoch sind beide Vorgehensweisen Alfresco-spezifisch, was somit eine Anbindung eines Clients an andere ECM-Systeme nicht möglich macht.

¹⁷ Dokumente, die nicht im SAP-System vorhanden sind, sondern anderen Systemen oder dem Benutzer als Datei vorliegen.

¹⁸ Eine Schnittstelle, welche es erlaubt verteilten Anwendungen miteinander zu kommunizieren.

¹⁹ Uniform Resource Identifier (engl. für „einheitlicher Bezeichner einer Ressource“).

- Die Alfresco Java Content Repository (JCR) RMI²⁰ Erweiterung überführt die JCR Implementierung in eine Anwendungsschicht, welche einen entfernten Zugriff gewährleisten soll und sich somit für die Integration von anderen Java Systemen qualifiziert [16]. Zwar handelt es sich bei dem Java Content Repository um einen industriellen Standard, doch der Einsatz dieser Schnittstelle ist oftmals an den Zweck des Repositoriums gebunden und setzt die Programmiersprache Java voraus.
- Die CMIS Spezifikation, ebenfalls von Alfresco unterstützt, bietet eine web-basierte Schnittstelle, die den Austausch von Inhalten jeglicher Art und den dazugehörigen Metadaten mit Hilfe einer standardisierten API ermöglicht [17].

Vergleicht man alle Herangehensweisen miteinander, so lässt sich erkennen, dass sich die Content-Management-Interoperability-Services am meisten für den Einsatz mit verschiedenen Repositorien eignen, da sie die Vorteile der Sprachunabhängigkeit und einer standardisierten Schnittstelle in sich vereinen. Aufgrund der Tatsache, dass jedes Unternehmen unterschiedliche Anforderungen an ein Enterprise-Content-Management-System stellt und die Schwerpunkte der einzelnen Lösungen doch gänzlich verschieden sein können, ist Einheitlichkeit ein wesentliches Kriterium bei der Auswahl der Schnittstelle.

6.1.4. Produktübersicht / Architektur

Das Produkt beschreibt eine Schnittstelle zum Archivieren von Dokumenten und den dazugehörigen Metadaten, welche Ergebnisse von Geschäftsprozessen einer SAP Unternehmenslösung sind. Die hauptsächlichen Anwendungsfälle des Connectors sind das Durchführen von Archivierungsaufgaben und das Verwalten von bereits archivierten Dokumenten. Dies beinhaltet alle Aufgaben eines SAP Content Servers. Des Weiteren soll das Produkt Möglichkeiten aufweisen, um administrative Tätigkeiten wie Überwachung, Protokollierung oder Konfiguration durchzuführen.

Die Interaktion mit dem Connector erfolgt über eine standardisierte HTTP Schnittstelle. Sie setzt somit die Generierung sowie das Versenden von Anfragen und Antworten, welche vom jeweiligen System verarbeitet werden, voraus.

²⁰ Remote Method Invocation (engl. für „Aufruf entfernter Methoden“)

Das folgende Komponentendiagramm soll die Ansiedlung des zu entwickelnden Softwaresystems zwischen den angrenzenden Applikationen zeigen.

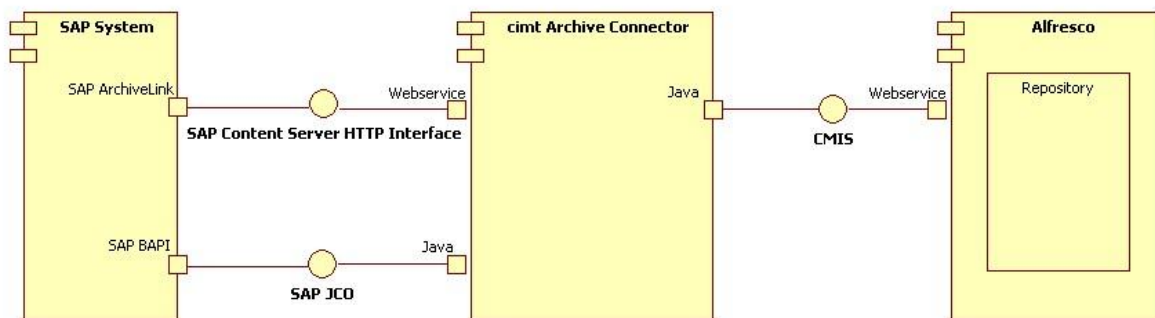


Abbildung 11: Komponentendiagramm: Architektur des Connectors

Es wird deutlich, dass der Connector als Vermittler zwischen den beiden angrenzenden Systemen agiert und auf Basis der vom SAP-System empfangenen Anfragen Manipulationen des Inhaltes durch die bereitgestellten Repository-Services vornimmt, Antworten generiert sowie zurücksendet. Mit Hilfe der optionalen SAP JCO Verbindung ist es möglich den Content mit zusätzlichen Metadaten anzureichern.

6.1.5. Anwendungsfall „Dokument erzeugen“

Um einen Eindruck von der Arbeitsweise des Connectors zu bekommen, soll in jeder Entwicklungsphase anhand der Pflichtenheftfunktion „Dokument erzeugen“ der Weg von einer abstrakten Anforderung bis hin zur konkreten Realisierung verfolgt werden.

In der folgenden Abbildung sieht man ein Anwendungsfalldiagramm, welches die erforderlichen Schritte beim Erzeugen eines Dokumentes zeigt. Im Rahmen eines dokumentenlastigen Geschäftsprozesses ist dort vom Nutzer eine Aufgabe in SAP auszuführen. Diese beinhaltet das Archivieren von Dokumenten und steht möglicherweise am Ende eines Workflows. Ein bereits zuvor erfolgtes Customizing, welches den SAP-Alfresco-Connector als Content-Server sowie Alfresco als Repository für die bei diesem Prozess entstehende Dokumentenart definiert hat, ermöglicht eine automatische HTTP-basierte Übertragung der abzulegenden Daten. Auf Seiten des späteren Softwareproduktes ist die empfangene Anfrage auszuwerten und in Folge dessen ein neues Dokument in Alfresco anzulegen. Dies kann, muss jedoch nicht, das Erzeugen ein oder mehrerer Komponenten bzw. das Erlangen von Metadaten beinhalten. Durch das Antworten mit dem entsprechenden HTTP-

Statuscode wird dem SAP-System ein Ergebnis geliefert, was sich in einem Erfolg oder einem Fehlschlag äußert.

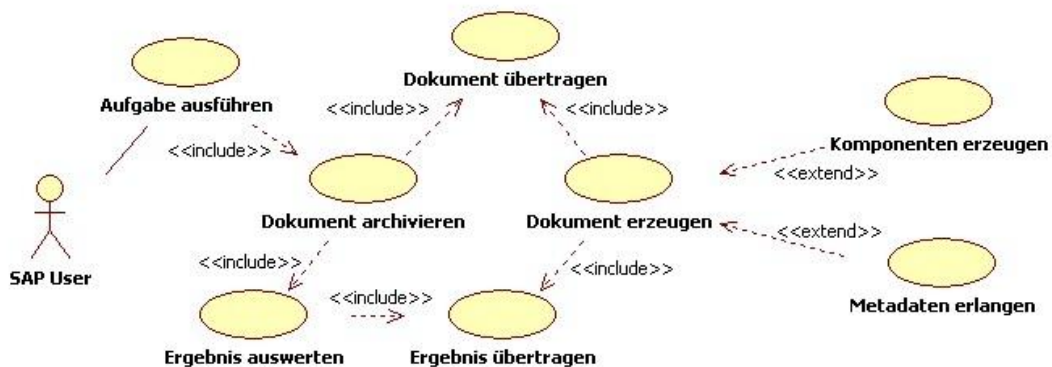


Abbildung 12: Anwendungsfalldiagramm: Dokument erzeugen

6.2. Entwurf

In der Phase des Entwurfs sind die einzelnen Systemkomponenten, unter Berücksichtigung der Zielplattform und den technischen Randbedingungen, zu spezifizieren. Dieser Abschnitt soll deshalb mit Hilfe von ausgewählten Komponenten das Design des Connectors verdeutlichen.

6.2.1. Charakteristik des Connectors

Aufgrund der Tatsache, dass die Einflussfaktoren des SAP-Alfresco-Connectors bereits in der Analysephase detailliert durch Festlegung des Produktzwecks und der Einsatzbedingungen spezifiziert wurden, sind an dieser Stelle nur noch wenige grundlegende Entscheidungen zu treffen.

Da das spätere Softwareprodukt eigenständig, d.h. ohne Beaufsichtigung eines Benutzers, eingesetzt werden soll, können die Benutzeranforderungen, mit Ausnahme der administrativen Funktionalitäten, vernachlässigt werden. Der Connector muss eine parallele Verarbeitung von Anfragen (HTTP-Requests) sowie einen sehr hohen Verfügbarkeitsgrad gewährleisten. Dies ist der Mehrbenutzerfähigkeit des angrenzenden SAP-Systems zu schulden.

Wie bereits in den Kriterien (siehe **Tabelle 4**) konkretisiert, muss die Realisierung der Schnittstelle innerhalb von Java Servlet Container lauffähig sein und auf der Java Enterprise Edition 6 basieren.

Die Grundsatzentscheidung bezüglich der Benutzeroberfläche kann ebenfalls stark vernachlässigt werden, da das Produkt selbst keine GUI besitzen soll. Bei einer JMX-basierten Konfiguration werden lediglich die Einstellungsmöglichkeiten definiert, jedoch nicht die Art oder das Aussehen der im JXM-Werkzeug verwendeten Oberfläche.

6.2.2. Architekturmodell

In *Abbildung 11* erkennt man, dass eine Integration des Connector in ein existierendes SAP-System und die Nutzung eines CMIS-konformen ECM-Systems, in diesem Fall Alfresco, vorgesehen ist. Daraus resultiert eine Verteilung der einzelnen Softwareschichten im Netz. Ähnlich wie bei der Architektur eines Enterprise-Content-Management-Systems kann auch hier, betrachtet man die gesamte Systemlandschaft und einen beliebigen Hauptanwendungsfall, eine Strukturierung in fünf Schichten vorgenommen werden. Dabei sind Präsentation und Steuerung ausschließlich auf Seiten des ERP Systems zu finden, in welchem die Archivierungsaufgaben anfallen. Die Zuordnung der Anwendungsebene gestaltet sich dagegen schwierig, da diese verteilt in allen drei Softwaresystemen vorliegt. Eine Verbindung zwischen den einzelnen Anwendungsbereichen erfolgt in der Regel nur temporär auf Anfrage, was wiederum charakteristisch für eine serviceorientierte Architektur ist. Hierbei übernimmt der SAP-Alfresco-Connector sowohl die Rolle eines Servers als auch die eines Clients, weil das SAP-System seine Content Server Dienste nutzt und er selbst die Web-Services von Alfresco. Die Enterprise-Content-Management Lösung ist letztendlich für den Datenzugriff und die Datenhaltung zuständig.

6.2.3. Verwaltung von Content-Repositories

Einen wesentlichen Aspekt bei der Nutzung des Connectors von SAP ArchiveLink beinhaltet die eindeutige Abbildung von Content-Repositories innerhalb des Softwaresystems, da diese beim Wiederauffinden eines abgelegten Dokumentes den Speicherraum, also das angebundene CMS, identifizieren. Die Grundlage für eine Anbindung eines CMIS-fähigen Repositoriums stellt die Klasse **RepositoryParameters** dar, welche die üblichen Sitzungsparameter wie Benutzername und Kennwort um ein weiteres Attribut für die in SAP vergebende zweistellige Content-Repository ID erweitert. Sofern vor dem Starten des Connectors

konfiguriert, werden ein oder mehrere solcher Instanzen der **RepositoryFactory** in Form einer List übergeben. Durch die Iteration der Objektliste wird mit je einem Zusammenschluss von Parametern ein neues **Repository** erzeugt, so dass jedes Repositorium eine eigene CMIS-Sitzung für die Interaktion mit dem Ablageort besitzt. Im Anschluss wird eine Assoziation zwischen Repositorium und der eindeutigen ID hergestellt, um anhand dieser den erforderlichen Verwaltungsort ansprechen zu können.

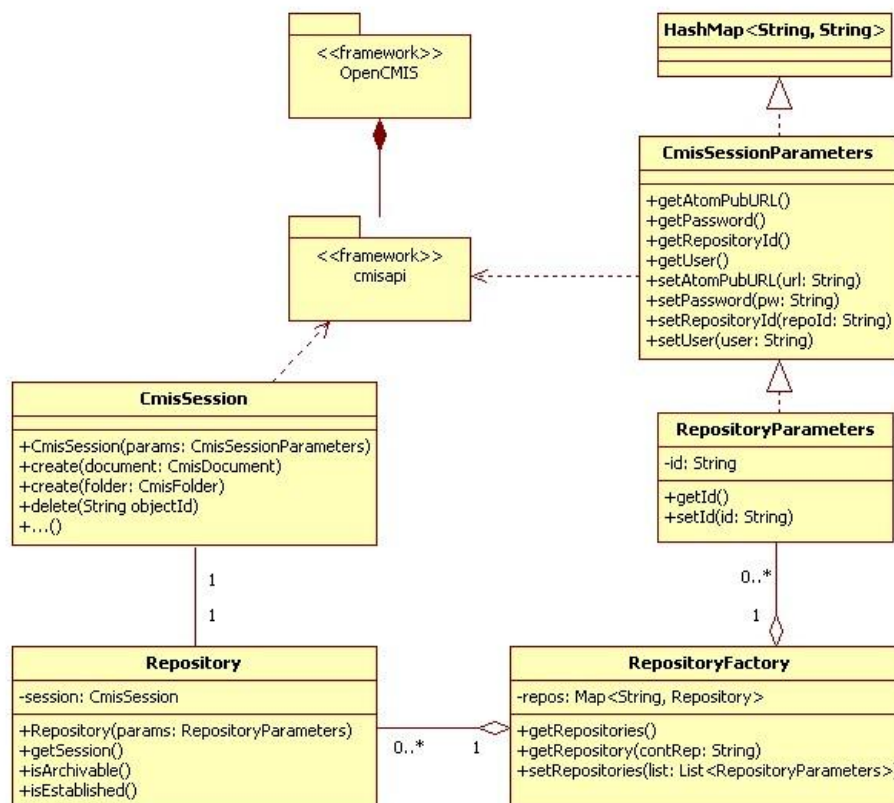


Abbildung 13: Klassendiagramm: Content-Repository

6.2.4. Umsetzung des SAP-internen Dokumentenbegriffs

Um Archivierungsaufgaben durchführen zu können, ist es erforderlich die Struktur eines Dokuments (vgl. 5.5.1) flexibel genug abzubilden, damit alle auftretenden Sonderfälle z.B. ein Dokument ohne Komponenten berücksichtigt werden können. In diesem Zusammenhang spielt einerseits der Entwurf eines Content-Modells und andererseits die Möglichkeit mit Hilfe des Content-Management-Interoperability Standards individuelle Datentypen zu verarbeiten eine entscheidende Rolle. Da die

eingesetzte CMIS Client API Apache Chemistry OpenCMIS²¹ die konkrete Implementierung der Klassen hinter einer Reihe von Programmierschnittstellen verbirgt, bietet sich im Gegensatz zum grundlegenden Konzept der Vererbung als einzige brauchbare Lösung sogenanntes Wrapping an. Dabei werden die, durch die CMIS-Sitzung eines Repositoriums erzeugten Objekte, durch Instanzen von **ArchiveLinkSystem**, **ArchiveLinkDocument** oder **ArchiveLinkComponent** umhüllt, so dass ihre Attribute und Methoden innerhalb der jeweiligen Klasse global verfügbar werden. Die folgende Abbildung zeigt ein UML-Klassendiagramm, welches den gewählten Ansatz verdeutlichen soll.

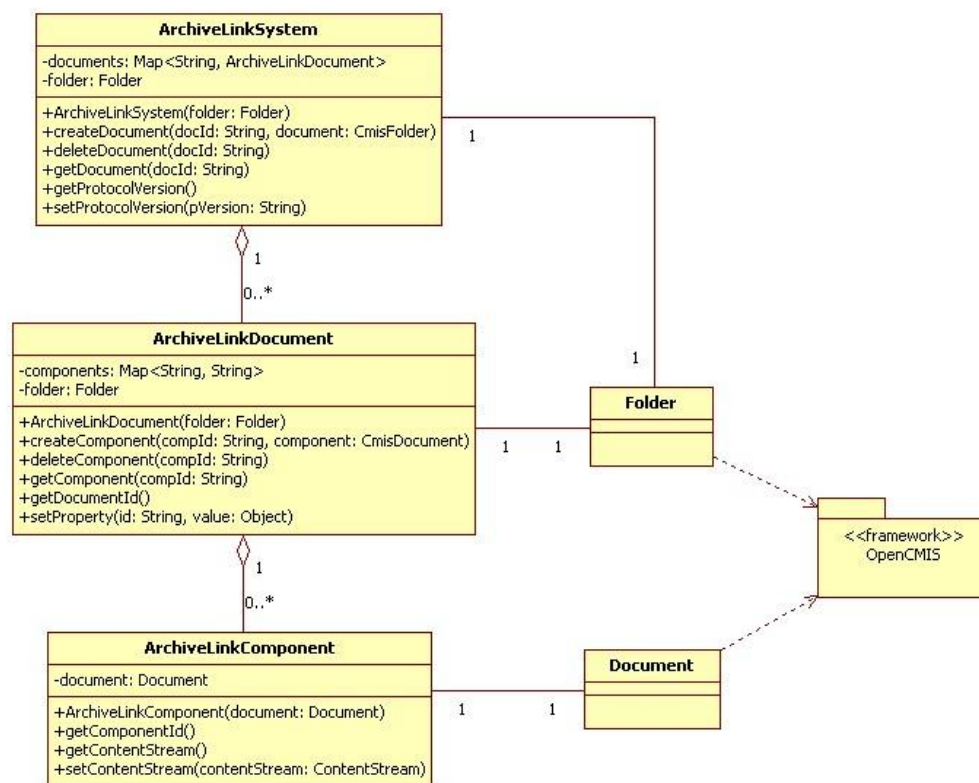


Abbildung 14: Klassendiagramm: Content-Modell

Obwohl die Klasse **ArchiveLinkDocument** ein abzulegendes Dokument repräsentiert, wird sie als Ordner realisiert, da ein Dokument der Definition nach auch mehrere Komponenten umfassen kann. Ein weiterer wichtiger Punkt ist das Wiederauffinden von Dokumenten und deren Komponenten. Die Schwierigkeit darin besteht die Anfrage des Nutzers in möglichst kurzer Zeit zu beantworten, so dass

²¹ Ein Java-basiertes Framework zur Nutzung der CMIS-Spezifikation [20].

dieser vorgangsbezogene Objekte im SAP-System mit nur geringer Verzögerung angezeigt bekommt. Um dies zu ermöglichen, werden Beziehungen zwischen Dokument- und Objekt-ID sowie zwischen Komponenten- und Objekt-ID innerhalb eines assoziativen Speichers gehalten. Ist ein abgelegtes Dokument nicht gespeichert, so wird eine Abfrage über das gesamte Content-Repository gestartet. Damit der Suchaufwand auch bei großen Datenmengen keine Performance-Probleme verursacht, bietet Alfresco die Möglichkeit innerhalb eines Content-Modells ausgewählte Attribute für eine Indizierung²² vorzuschlagen.

6.2.5. Sequenz „Dokument erzeugen“

Zur genaueren Modellierung einer Funktion bietet sich in der Regel das Sequenzdiagramm an. Damit lässt sich eine bestimmte Sicht generieren, welche die Interaktionen zwischen sowie die Lebenszyklen von Objekten abbildet und somit dynamische Aspekte für eine leichtere Realisierung vermittelt.

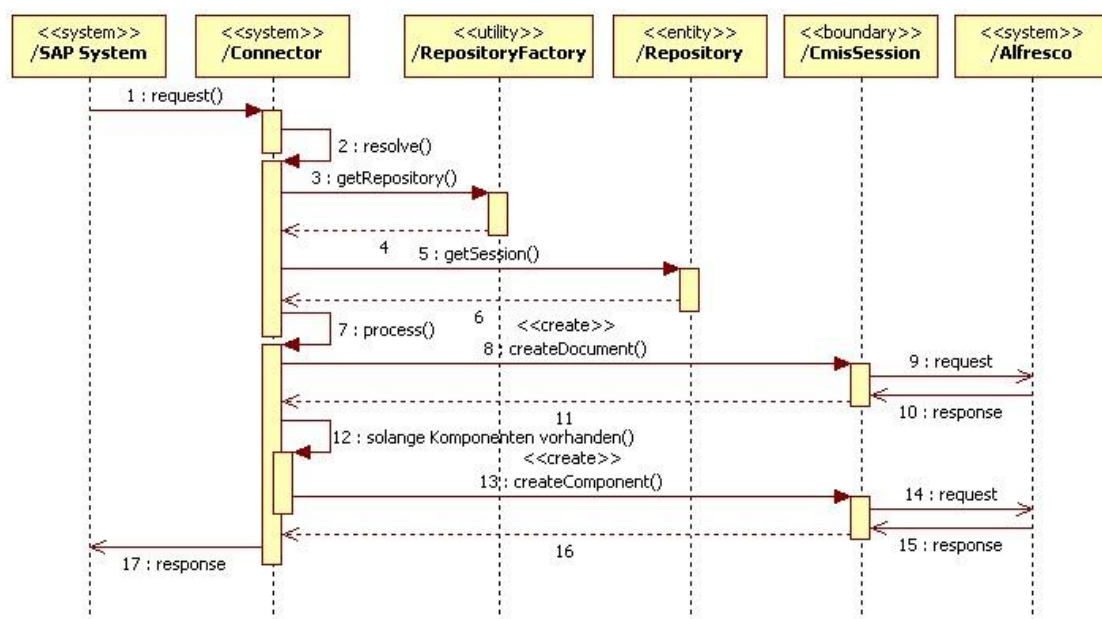


Abbildung 15: Sequenzdiagramm: Dokument erzeugen

Um die Übersichtlichkeit zu bewahren, sind im abgebildeten Diagramm einige Abläufe lediglich zeitlich eingeordnet und nicht detailliert aufgeschlüsselt. So löst Zweitens die Anfrage des SAP-Systems auf, indem den richtigen Verarbeitungs-

²² Ein Verfahren zur Beschleunigung von häufig gestellten Anfragen an eine Datenbank.

prozess die notwendigen Parameter und Daten zur Verfügung gestellt werden. Im nächsten Schritt wird mit Hilfe der Content-Repository-ID das entsprechende Repositorium einschließlich der zur Interaktion mit Alfresco benötigten CMIS-Sitzung erlangt. Unter dem siebten Punkt findet die Verarbeitung statt, was neben der Trennung des durch die Anfrage mitgelieferten Datenstroms in einzelne Komponenten die Gewinnung von Metadaten für das Erstellen der Objekte wie z.B. Dateiname, -länge, Dokumentenart und Medientyp beinhaltet. Im Anschluss wird dann das Dokument mit seinen Komponenten im Repositorium erzeugt und eine Antwort generiert, damit SAP entsprechend des Ergebnisses reagieren kann.

6.3. *Implementierung*

Die Implementierungsphase beschreibt die konkrete Realisierung des Entwurfsmodells. Im Rahmen dieses Entwicklungsprozesses soll anhand von Codeauszügen und konkreten Ablaufbeschreibungen eine Lösung der Problemstellung aufgezeigt werden. Die Verwendung verschiedener Frameworks stellt dabei eine erhebliche Entlastung, vor allem bei der Implementierung der Kommunikationsschnittstellen, dar, so dass eine Fokussierung auf geforderte Qualitätsmerkmale wie Funktionalität, Zuverlässigkeit und Effizienz erfolgen kann.

6.3.1. Einsatz von JAX-RS

Die *Java API for RESTful Web Services (JAX-RS)* beschreibt eine Spezifikation zur Unterstützung der Realisierung von verteilten Informationssystemen mit Hilfe Softwarearchitekturstils „Representational State Transfer“ [18]. Durch die Verwendung von Java Annotierungen können den Skriptpfaden bestimmte Klassen und den HTTP-Kommandos auszuführende Methoden zugewiesen werden. Dies hat zur Folge, dass die Verteilung der Anfragen sowie das Auflösen der Parameter automatisch geschehen und somit eine enorme Erleichterung bei der Entwicklung darstellen. Um einen Einsatz von JAX-RS erst einmal zu ermöglichen, ist es notwendig die eingehenden Anfragen umzuleiten, so dass die resultierenden Request-URLs ein gewisses Schema erfüllen. Die ursprüngliche Syntax, welche im SAP Content Server HTTP Standard definiert ist, sieht wie folgt aus:

`http://<servername>:<port>/<script>?<command>&<parameters>`

Beispiel:

<http://localhost:8080/connector/archivelink?serverInfo&pVersion=0045>

Der **servername** beschreibt die Internetadresse des Servers, auf den der Zugriff über einen TCP/IP **port** stattfindet. Das **script** benennt das Programm, das die Funktionalitäten eines Content Servers implementiert. Durch das **command** wird die auszuführende Funktion, auf welche die einzelnen Parameter anzuwenden sind, identifiziert.

Damit eine Aufteilung der Gesamtverarbeitung in einzelne Ressourcen erfolgen kann, muss das Kommando in den Skriptpfad integriert werden. Dafür zuständig ist der **Redirector**, ein Servlet, welches mit Hilfe einer **HttpURLConnection** die Umleitung der Anfragen übernimmt. Innerhalb dieses Prozesses müssen neben der Generierung der neuen URL auch Request-Header und -Body auf die erstellte Verbindung übertragen werden. Ist dies geschehen, ergibt sich die folgende Adresse, unter welcher die für das Kommando zuständige Ressource zu finden ist:

<http://<servername>:<port>/<script>/<command>?<parameters>>

Beispiel:

<http://localhost:8080/connector/api/serverInfo?pVersion=0045>

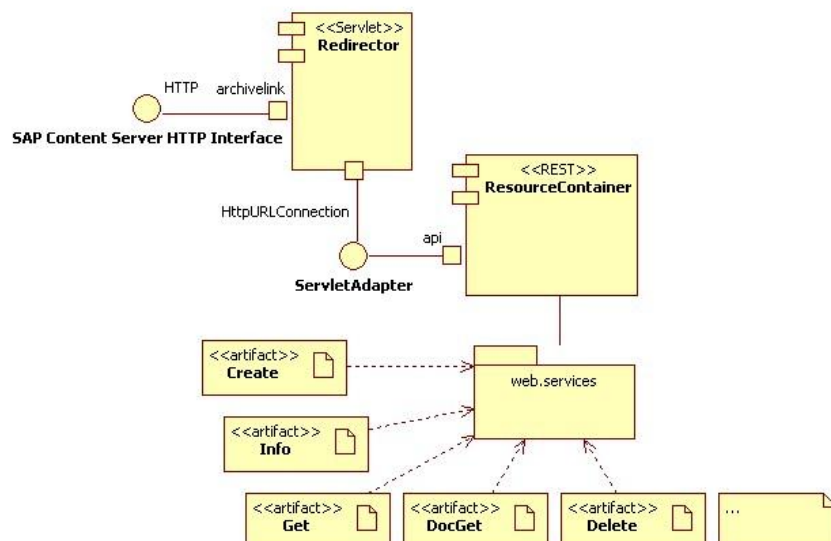


Abbildung 16: Komponentendiagramm: Redirection

6.3.2. Webservice „Dokument erzeugen“

Anhand eines Webservices ist nun zu zeigen wie JAX-RS eingesetzt wird und die konkrete Implementierung einer Funktion der HTTP-Schnittstelle aussieht.

Um den Dienst für das Erzeugen eines Dokumentes verfügbar zu machen, muss die Klasse durch die `@Path`-Annotierung markiert werden. Diese sorgt dafür, dass die betreffende Ressource beim Starten der Webanwendung registriert wird und nun vom `ServletAdapter` (siehe *Abbildung 16*) verwendet werden kann.

```
40    @Path(HttpCommands.CREATE)
41    public class Create implements HttpHeaders, HttpParameters {
```

Listing 1: Pfad-Annotierung

Ebenfalls die auszuführenden Methoden sind zu kennzeichnen, damit eine Verbindung zwischen Operation und HTTP-Kommando wie in diesem Fall `@Post` hergestellt werden kann. Die Auflösung der Parameter innerhalb einer Anfrage erfolgt wiederum mit Hilfe von Annotierungen, welche eine eindeutige Zuweisung garantieren.

```
74    @POST
75    public Response post(
76        @QueryParam(PROTOCOL_VERSION) String pVersion,
77        @QueryParam(CONTENT_REPOSITORY) String contRep,
78        @QueryParam(DOCUMENT_ID) String docId,
79        @QueryParam(DOCUMENT_PROTECTION) String docProt,
80        @QueryParam(ACCESS_MODE) String accessMode,
81        @QueryParam(AUTHENTICATION_ID) String authId,
82        @QueryParam(EXPIRATION) String expiration,
83        @QueryParam(SECURITY_KEY) String secKey,
84        @Context HttpHeaders headers,
85        InputStream body) {
```

Listing 2: Methodenkopf

Wie im Entwurf spezifiziert, ist als allererstes der Zugriff auf den Ablageort zu erlangen, welcher durch ein `Repository`-Objekt repräsentiert wird und einer Content-Repository-ID zugewiesen ist. Sollte kein Ablageort mit dem gegebenen Schlüssel existieren, so ist laut des SAP Content Server HTTP Standards wie bei jedem anderen auftretenden Fehler ein spezifischer Code zurückzugeben, der dem SAP-System das Problem verdeutlicht.

```

87         //Get repository
88         Repository repo = Application.getRepositoryFactory().getRepository(contRep);
89
90         //If repository not existing
91         //throw 'Bad Request (400)' exception
92         if(repo == null) {
93             throw new WebApplicationException(Response.Status.BAD_REQUEST);
94         }

```

Listing 3: Repositorium erlangen

Im folgenden Schritt werden aus den Daten des Request-Body's die benötigten Metadaten extrahiert und die Komponenten nach Vorgabe der Spezifikation für den Medientyp „multipart/form-data“ gelesen [19].

```

96         //Get multipart form data
97         MultipartFormDataInput multipart = null;
98         try {
99             //Create multipart form data
100             multipart = new MultipartFormDataInput(headers.getRequestHeader(CONTENT_TYPE).get(0), body);
101         } catch(IllegalArgumentException e) {
102             //If content type is wrong
103             //throw 'Bad Request (400)' exception
104             throw new WebApplicationException(e, Response.Status.BAD_REQUEST);
105         } catch(IOException e) {
106             //If form data not created
107             //throw 'Internal Server Error (500)' exception
108             throw new WebApplicationException(e, Response.Status.INTERNAL_SERVER_ERROR);
109         }

```

Listing 4: Multipart Form-Data erzeugen

Ist dies geschehen, so wird ein Dokument als Container für die einzelnen Komponenten erstellt. Da der Name eindeutig sein muss, wird in diesem Fall die von SAP vergebene Dokumenten-ID benutzt.

```

117         //Create document
118         CmisFolder folder = new CmisFolder(docId, ArchiveLinkDocument.getObjectTypeId());
119
120         //Try to create within repository
121         try {
122             //Create
123             repo.getArchiveSystem().createDocument(docId, folder);
124         } catch(CmisNameConstraintViolationException e) {
125             //If document already exists
126             //throw 'Forbidden (403)' exception
127             throw new WebApplicationException(e, Response.Status.FORBIDDEN);
128         } catch(CmisPermissionDeniedException e) {
129             //If create permission missing
130             //throw 'Unauthorized (401)' exception
131             throw new WebApplicationException(e, Response.Status.UNAUTHORIZED);
132         } catch(CmisStorageException e) {
133             //If document not stored
134             //throw 'Internal Server Error (500)' exception
135             throw new WebApplicationException(e, Response.Status.INTERNAL_SERVER_ERROR);
136         }

```

Listing 5: Dokument erstellen

Danach werden die Komponenten erstellt, indem die ermittelten Metadaten zum Beschreiben des jeweiligen Datenstroms benutzt werden. Sind alle notwendigen Merkmale charakterisiert, so kann die Komponente dem Dokument hinzugefügt werden.

```
138 //Get created document
139 ArchiveLinkDocument document = repo.getArchiveSystem().getDocument(docId);
140
141 //Iterate form data
142 for(FormDataInput formDataInput : multipart.getFormData()) {
143     //Get metadata
144     String mimeType = formDataInput.getContentType();
145     String filename = formDataInput.getFilename() + Application.getMimeTypeMapper().getExtension(mimeType);
146     String compId = formDataInput.getComponentId();
147
148     //Create content stream
149     CmisContentStream contentStream = new CmisContentStream();
150     contentStream.setFileName(filename);
151     contentStream.setLength(BigInteger.valueOf(formDataInput.getContentLength()));
152     contentStream.setMimeType(mimeType);
153     contentStream.setStream(formDataInput.getData());
154
155     //Create component
156     CmisDocument component = new CmisDocument(filename, ArchiveLinkComponent.getObjectTypeId());
157     component.setContentStream(contentStream);
158
159     //Try to create within repository
160     try {
161         //Create
162         document.createComponent(compId, component);
163     }
```

Listing 6: Komponenten erstellen

Als letztes wird eine Antwort generiert, welche das erfolgreiche Ablegen eines Dokumentes in dem externen Speichermedium belegt.

```
205 //Create task for specifying document
206 Application.getTaskManager().getTaskPool().add(new SpecifyDocument(contRep, docId));
207
208 //Return 'Created {201}'
209 return Response.status(Response.Status.CREATED).build();
210 }
```

Listing 7: Antwort generieren

6.3.3. Archive-Type-Mapping

Resultierend aus der detaillierten Planung des Softwaresystems, konnte die Implementierung früher als erwartet abgeschlossen werden, so dass noch Kapazitäten für das Umsetzen eines weiteren wichtigen Produktmerkmals zur Verfügung standen. Durch den Einsatz des SAP Java Connectors sollen die abgelegten Dokumente mit den Metadaten der dazugehörigen Business-Objekte angereichert werden. Um dies zu erreichen, ist die Idee eines generischen Mappings entstanden, welches SAP-Objektypen mit CMIS-Datentypen verknüpft und anhand

von festgelegten Vorgehensweisen das Erlangen der gewünschten Informationen spezifiziert.

Die XML-Technologie stellt hierbei die Grundlage für eine Realisierung dar, da diese das Modellieren eigener Datenstrukturen sowohl händig als auch maschinell gewährleistet. Mit dem Definieren eines Schemas wird nicht nur die Validierung des Archive-Type-Mappings möglich, sondern auch die Verwendung des JAXB-Frameworks. Dieses beschreibt eine Programmierschnittstelle, welche aus einem XML-Schema Java-Klassen generiert und durch Datenbindung eine Verknüpfung zwischen selbigen herstellt. Die Vorteile für den Entwickler liegen darin, dass er sich weder mit Serialisierung, noch mit Deserialisierung ferner mit Schnittstellen zur Verarbeitung von XML wie SAX²³ oder DOM²⁴ beschäftigen muss, da diese Aufgaben von JAXB übernommen werden.

Im folgenden Abschnitt soll der Aufbau einer solchen Mapping-Datei am Beispiel des Objekttyps „Verkaufsbelegrechnungskopf“ (VBRK) gezeigt werden, um die Funktionsweise und die notwendigen Schritte beim Erlangen der Metadaten aus SAP zu verstehen.

Die zum Verknüpfen verwendeten Content-Typen werden in der XML-Datei, wie in **Listing 8** zu sehen, abgebildet. Anhand der *Objektyp-ID* erfolgt eine eindeutige Identifikation des Typs, welche die Voraussetzung für das spätere Mapping mit einem SAP-Objektyp darstellt. Diese Herangehensweise schließt Namenskonflikte aus und wird daher ebenso beim Erstellen der Merkmalsbeziehungen genutzt. Da der Content-Management-Interoperability-Services Standard die Vererbung von Merkmalen vorsieht, wird mit Hilfe der *hierarchischen Beziehung* ein Bezug zu den generelleren Typen hergestellt.

²³ SAX steht für „Simple API for XML“ und beschreibt eine Programmierschnittstelle zum Verarbeiten von XML-Dateien in Java.

²⁴ Das „Document Object Model“ spezifiziert eine Schnittstelle, welche einen dynamischen Zugriff auf HTML- und XML-Dokumente ermöglicht.

```

<cmis-type id="F:archivelink:billingDocument">
  <name>Billing Document</name>
  <description>A billing document</description>
  <parent>archivelink:document</parent>
  <properties>
    <property name="archivelink:billType">
      <name>Billing Type</name>
      <description>Fakturaart</description>
      <type>java.lang.String</type>
    </property>
    <property name="archivelink:currency">
      <name>Currency</name>
      <description>Waehrung des Vertriebsbelegs</description>
      <type>java.lang.String</type>
    </property>
  </properties>

```

Objekttyp-ID

Hierarchische Beziehung

Merkmals-ID

Datentyp des Merkmals

Listing 8: CMIS-Typdefinition

Ein Verfahren um Informationen über ein bestimmtes Geschäftsobjekt zu erlangen, ist der Aufruf eines Funktionsbausteins, welcher durch einen technischen Namen innerhalb des SAP-Systems ansprechbar wird. Die *Eingangsdaten* beschreiben eine Reihe von Parametern. Diese sind vor dem eigentlichen Aufruf zu füllen und beeinflussen das Funktionsergebnis. Damit die ausgehenden Datenfelder exakt bestimmt werden können, sind eindeutige Schlüssel aus dem SAP-Objekttyp, den *Namen des Funktionsbausteins* und dem Feldnamen zu bilden.

```

<function-module name="BAPI_BILLINGDOC_GETDETAIL">
  <import-parameters>
    <simple-import>
      <name>BILLINGDOCUMENT</name>
      <type>java.lang.String</type>
      <value>OBJECT_ID</value>
    </simple-import>
  </import-parameters>
  <export-parameters>
    <complex-export>
      <name>BILLINGDOCUMENTDETAIL</name>
      <type>com.sap.conn.jco.JCoStructure</type>
      <properties>
        <property id="VBRK:BAPI_BILLINGDOC_GETDETAIL:BILL_TYPE">
          <name>BILL_TYPE</name>
          <description>Fakturaart</description>
          <type>java.lang.String</type>
        </property>
        <property id="VBRK:BAPI_BILLINGDOC_GETDETAIL:CURRENCY">
          <name>CURRENCY</name>
          <description>Waehrung des Vertriebsbelegs</description>
          <type>java.lang.String</type>
        </property>
      </properties>
    </complex-export>
  </export-parameters>

```

Name des Funktionsbausteins

Eingangsdaten

Technischer Name

Datentyp

Ausgangsdatenfeld

Listing 9: SAP-Funktionsbaustein

Aufgrund der Tatsache, dass nicht für jeden Objekttypen in SAP Funktionsbausteine existieren, stellt das Durchführen von Abfragen auf interne Systemtabellen eine

weitere Praktik zum Erlangen von Informationen dar. Um die Tabelle zu bestimmen wird lediglich der Tabellename angegeben. Wie in SQL bietet SAP gleichermaßen die Möglichkeit die Ergebnismenge durch den Einsatz von Operatoren wie z.B. „ist gleich“ oder „größer als“ einzuschränken. Auch hier sind die gewünschten Ausgangsfelder zu definieren und der einzigartige Identifikationsschlüssel zu generieren.

```

<query table="VBRK">
  <query-parameters>
    <options>
      <equals field="VBELN">
        <value>OBJECT_ID</value>
      </equals>
    </options>
    <rows>
      <count>1</count>
      <multiple>false</multiple>
    </rows>
  </query-parameters>
  <query-results>
    <query-result>
      <id>VBRK:VBRK:ERNAM</id>
      <name>ERNAM</name>
      <description>Sachbearbeiter</description>
      <type>C</type>
    </query-result>
  </query-results>
</query>

```

Tabellename

Datensatzfilter

Anzahl der zu verarbeitenden Zeilen

Abfrageergebnis

Listing 10: SAP-Tabellenabfrage

Als letztes werden die einzelnen Typen und deren Merkmale miteinander verknüpft, so dass nach dem Ausführen der, zum SAP-Objektypen dazugehörigen, Funktionen und Abfragen die definierten Merkmale Verwendung finden.

```

<cmis-type>F:archivelink:billingDocument</cmis-type>
<sap-type>VBRK</sap-type>
<property-mappings>
  <property-mapping>
    <cmis-property>cmis:name</cmis-property>
    <sap-property>VBRK:BAPI_BILLINGDOC_GETDETAIL:BILLINGDOC</sap-property>
  </property-mapping>
  <property-mapping>
    <cmis-property>archivelink:objId</cmis-property>
    <sap-property>VBRK:BAPI_BILLINGDOC_GETDETAIL:BILLINGDOC</sap-property>
  </property-mapping>
  <property-mapping>
    <cmis-property>archivelink:currency</cmis-property>
    <sap-property>VBRK:BAPI_BILLINGDOC_GETDETAIL:CURRENCY</sap-property>
  </property-mapping>
</property-mappings>

```

Typverknüpfung

Merkmalsverknüpfung

Listing 11: Type-Mapping

6.3.4. Aufgabe „Dokument spezifizieren“

Der Eintrag in der Verknüpfungstabelle repräsentiert für jedes abgelegte Dokument den Ausgangspunkt beim Erlangen zusätzlicher Metadaten, weil ausschließlich an dieser Stelle eine Beziehung zwischen Dokument-ID, Geschäftsobjekt-ID und Objekttypen zu finden ist (vgl. *Abbildung 9*). Die Vorgehensweise beim Ablegen eines Dokumentes sieht vor, dass die Verknüpfung erst dann geschrieben wird, wenn das Dokument erfolgreich im Content-Repository abgelegt wurde. Dadurch musste sich eine Prozedur überlegt werden, die das Spezifizieren eines Dokumentes möglich zeitnah nach dem Erstellen gewährleistet.

Direkt nach dem Erstellen des Dokumentes wird daher eine neue Spezifikationsaufgabe erzeugt, welche von einem periodisch ablaufenden Prozess ausgeführt wird. Sobald ein passender Eintrag in der Verknüpfungstabelle vorhanden ist, wird anhand der Objekttyp-ID das Archive-Type-Mapping nach entsprechenden Funktions- und Abfragedefinitionen durchsucht, welche im Anschluss mit den Ausgangsdaten verarbeitet werden. Ist dies geschehen, so wird ein spezielles Dokument mit den Metadaten des dazugehörigen Geschäftsobjektes erstellt, die Komponenten umgehungen bzw. verschoben wie man es aus herkömmlichen Dateisystemen kennt, und das ursprüngliche Dokument gelöscht.

7. Zusammenfassung

Das letzte Kapitel soll eine Darstellung der erzielten Ergebnisse sowie eine Wertung dieser Arbeit liefern.

7.1. *Fazit*

Durch diese Arbeit konnte bewiesen werden, dass eine Integration von Enterprise-Content-Management-Systemen in vorhandene IT-Landschaften möglich ist und diese eine geeignete Strategie beschreibt Informationen unternehmensweit verfügbar zu machen. Damit Individualentwicklungen und die Verwendung von Open Source Produkten in diesem Bezug die genannten Vorteile mit sich bringen, sind die konkreten Einsatzmöglichkeiten mit Hilfe einer detaillierten Anforderungsplanung zu identifizieren. Bei der Konzipierung und Realisierung ist es dann nicht nur erforderlich die Prinzipien, Methoden und Werkzeuge einer ingenieurmäßigen Entwicklung von Softwaresystemen zu verwenden, sondern auch auf standardisierte Vorgehensweisen zu setzen, welche die Lebensdauer von Software drastisch erhöhen. So stellen einheitliche Schnittstellen die Grundvoraussetzung für die Austauschbarkeit von Produkten dar, was vor allem bei Unternehmen mit sich häufig ändernden Anforderungen oder IT-Dienstleistern mit Kunden aus unterschiedlichen Sektoren einen signifikanten Wettbewerbsvorteil zur Folge hat. Letztendlich ist festzuhalten, dass zwar verbreitete Anwendungsfälle wie die Archivierung existieren. Jedoch die jeweils geltenden Anforderungen und Rahmenbedingungen stark voneinander abweichen und somit eine Evaluierung von Lösungsmöglichkeiten immer wieder notwendig ist.

Abschließend kann ich behaupten, dass die gesteckten Ziele erreicht worden und der SAP-Alfresco-Connector bereits in diesem Entwicklungsstand eine verwendbare Softwarekomponente darstellt, welche sich an individuelle Bedürfnisse anpassen lässt ohne Änderungen am Quellcode vornehmen zu müssen.

7.2. *Ausblicke*

Ein sich anschließendes Projekt könnte die vollständige Implementierung der SAP Content Server HTTP Schnittstelle anstreben, um eine Zertifizierung des Softwareproduktes durch die SAP AG zu erhalten. Des Weiteren wäre es denkbar Funktionalitäten auf Seiten von Alfresco zu realisieren, die Enterprise-Content-

Management mit SAP noch effektiver gestalten. So könnte beispielsweise „Frühe Archivierung“ und die Verknüpfung von Dokumenten mit SAP-Geschäftsobjekten aus Alfresco heraus ein Thema sein oder Mechanismen zur Aktualisierung von Metadaten der in Alfresco abgelegten Dokumente entwickelt werden.

Als einen festen Bestandteil der Weiterentwicklung sehe ich ebenfalls die Realisierung eines Werkzeuges zum Erstellen und Bearbeiten der XML-Mapping-Dateien. Dieses kann auf Basis der bereits eingesetzten Kommunikationsschnittstellen CMIS und SAP JCO beruhen, da beide Methoden zum Erlangen von Metadaten zu den existierenden Datenstrukturen bzw. Funktionsbausteinen bereitstellen.

8. Literaturverzeichnis

- [1] **AIIM.** Association for Information and Image Management. [Online] 2005.
<http://www.aiim.org/>.
- [2] **Caruana, David; Newton, John und Farman; Michael, Uzquiano; Michael G.; Roast, Kevin.** *Professional Alfresco - Practical Solutions for Enterprise Content Management*. Indianapolis : Wiley Publishing, Inc., 2010.
- [3] **Alfresco.** Alfresco Software Company Information. [Online] 2010.
<http://www.alfresco.com/about/>.
- [4] **Alfresco.** *Total Cost of Ownership for ECM*. [White Paper] Berkshire : Alfresco Software Inc., 2010.
- [5] **Kaib, Michael.** *Enterprise Application Integration: Grundlagen, Integrationsprodukte, Anwendungsbeispiele*. Marburg : Deutscher Universitäts-Verlag/GWV Fachverlage GmbH, 2002.
- [6] **Horn, Torsten.** EAI Enterprise Application Integration. [Online] 2007.
<http://www.torsten-horn.de/techdocs/eai.htm>.
- [7] **Heck, Rinaldo.** *Geschäftsprozessorientiertes Dokumentenmanagement mit SAP*. Bonn : Galileo Press, 2009.
- [8] **Müller, Joachim.** *Workflow based integration*. Berlin : Springer Verlag, 2005.
- [9] **Camphausen, Bernd.** *Grundlagen der Betriebswirtschaftslehre*. München : Oldenbourg Wissenschaftsverlag GmbH, 2008.
- [10] **Gutzmann, Ulrike.** *Arbeitskreis "Elektronische Archivierung"*. 2004-2007.
- [11] **Pledger, Tulin.** SAP.info. *Archivierungslösungen*. [Online] 24. März 2009.
<http://de.sap.info/archivierungslösungen/11693>.
- [12] **Sun Microsystems.** J2EE 1.4 Tutorial. *Chapter 1: Overview*. [Online] 2005.
<http://download.oracle.com/javaee/1.4/tutorial/doc/>.
- [13] **SAP AG.** *SAP ArchiveLink*. [Dokumentation] 2001.

- [14] **AlfrescoWiki.** Web Scripts. *Why Web Scripts?* [Online] 13. Mai 2010.
http://wiki.alfresco.com/wiki/HTTP_API#Why_Web_Scripts.3F.
- [15] **AlfrescoWiki.** Developer Guide. *Developing against the Alfresco Repository.*
[Online] 10. Juni 2010. http://wiki.alfresco.com/wiki/Developer_Guide#Developing_against_the_Alfresco_Repository.
- [16] **AlfrescoWiki.** Is JCR-RMI Right for Me? *Alfresco JCR RMI Extension.*
[Online] 03. Februar 2010. http://wiki.alfresco.com/wiki/Is_JCR-RMI_Right_for_Me%3F#Alfresco_JCR_RMI_Extention.
- [17] **ECM, IBM & Microsoft Corporation.** Vision / Mission. *Content Management Interoperability Services. Part I: Introduction, General Concepts, Data Model, and Services.* 2008. Version 0.5 (Seite 7).
- [18] **Java Community Process Program.** JSR 311: JAX-RS: The Java API for RESTful Web Services. *Section 2: Request.* [Online] 23. November 2009.
<http://jcp.org/en/jsr/detail?id=311>.
- [19] **Network Working Group.** Internet Engineering Task Force. *Returning Values from Forms: multipart/form-data.* [Online] August 1998.
<http://www.ietf.org/rfc/rfc2388.txt>.
- [20] **Apache Chemistry.** OpenCMIS. [Online] 2010.
<https://cwiki.apache.org/CMIS/opencmis.html>.
- [21] **Sun Microsystems.** Sun Developer Network (SDN). *JavaServer Pages Technology.* [Online] 2010. <http://java.sun.com/products/jsp/index.jsp>.
- [22] **W3C.** World Wide Web Consortium. [Online] 2010. <http://www.w3.org/>.
- [23] **Stegmann, Ute.** *Einsatz von Java-Frameworks in verteilten Systemen.* [Bachelorarbeit] Hamburg, 2005.

v. Erklärung zur selbstständigen Anfertigung

Erklärung:

Ich erkläre, dass ich die vorliegende Arbeit selbstständig und nur unter Verwendung der angegebenen Literatur und Hilfsmittel angefertigt habe.

Bearbeitungsort, Datum

Unterschrift